International Workshop on Spoken Language Translation

December 4–5, 2014

Proceedings

Lake Tahoe, CA

iwslt2014.org

Proceedings of the

International Workshop on Spoken Language Translation

December 4th and 5th 2014

Lake Tahoe, CA, USA

Edited by

Marcello Federico

Sebastian Stücker

François Yvon

Contents

Foreword

The International Workshop on Spoken Language Translation (IWSLT) is an annual scientific workshop, associated with an open evaluation campaign on Spoken Language Translation, where both scientific papers and system descriptions are presented. The 11th International Workshop on Spoken Language Translation takes place in Lake Tahoe, USA on Dec. 04 and 05, 2014. Since 2004, the annual workshop has been held in Kyoto, Pittsburgh, Kyoto, Trento, Honolulu, Tokyo, Paris, San Francisco, Hong Kong, and Heidelberg, and this year in Lake Tahoe.

One of the prominent research activities in Spoken Language Translation is the work conducted by the Consortium for Speech Translation Advanced Research (C-STAR), which was an international partnership of research laboratories engaged in automatic translation of spoken language started in early 90s. The C-STAR members had initiated the first shared task-type Spoken Language Translation workshop in 2004 and the IWSLT has been growing up with more participants and steering committee members.

The IWSLT includes scientific papers in dedicated technical sessions, either in oral or poster form. The contributions cover theoretical and practical issues in the field of Machine Translation (MT) in general and Spoken Language Translation (SLT), including Automatic Speech Recognition (ASR), Text-to-Speech Synthesis (TTS), and MT, in particular:

- Speech and text MT
- Integration of ASR and MT
- MT and SLT approaches
- MT and SLT evaluation
- Language resources for MT and SLT
- Open source software for MT and SLT
- Adaptation in MT
- Simultaneous speech translation
- Speech translation of lectures
- Efficiency in MT
- Stream-based algorithms for MT
- Multilingual ASR and TTS
- Rich transcription of speech for MT
- Translation of on-verbal events

Submitted manuscripts were carefully peer-reviewed by three members of the program committee and papers were selected based on their technical merit and relevance to the conference. In addition to core statistical machine translation papers, the technical program covers a wide spectrum of topics related to Spoken Language Translation,

ranging from issues related to real-time interpretation or to the translation of dialogs to more practical issues related to the integration of speech and translation technologies. Several important new annotated corpora will also be presented during the workshop. In summary, the large number of submissions as well as the high quality of the submitted papers indicates the interest on Spoken Language Translation as a research field and the growing interest in these technologies and their practical applications.

The results of the Spoken Language Translation evaluation campaigns organized in the framework of the workshop are also an important part of IWSLT. Those evaluations are organized in the manner of competition. While participants compete for achieving the best result in the evaluation, they come together afterwards, and discuss and share their techniques that they used in their systems. In this respect, IWSLT proposes challenging research tasks and an open experimental infrastructure for the scientific community working on spoken and written language translation. This year, the IWSLT evaluation offered a very challenging and appealing task on the Spoken Language Translation of public speeches (TALK) in a variety of topics, including a dedicated task to automatic speech recognition in order to cover the full pipeline of speech translation.

For each task, monolingual and bilingual language resources, as needed, are provided to participants in order to train their systems, as well as sets of manual and automatic speech transcripts (with n-best and lattices) and reference translations, allowing researchers working only on written language translation to also participate. Moreover, blind test sets are released and all translation outputs produced by the participants are evaluated using several automatic translation quality metrics. For the primary submissions of all MT and SLT tasks, a human evaluation was carried out as well.

Each participant in the evaluation campaign has been requested to submit a paper describing the system and the utilized resources. A survey of the evaluation campaigns is presented by the organizers.

This time IWSLT 2014 is co-located with the 2014 IEEE Spoken Language Technology Workshop (SLT 2014). SLT will be held in South Lake Tahoe, Nevada, on Dec. 7-10, 2014. The main theme of the SLT workshop will be "machine learning in spoken language technologies". We expect that the co-location of IWSLT 2014 and SLT 2014 will attract more participants for further discussion on multi-lingual spoken language technologies.

Apart from the technical content of the conference, spectacular scenery of Lake Tahoe will welcome all participants to around-the-clock awesomeness on the shore of the largest alpine lake in North America.

Welcome to Lake Tahoe!
Satoshi Nakamura, General Chair IWSLT 2014

Organizers

Chairs

Satoshi Nakamura (NAIST Japan): General Chair
Marcello Federico (FBK, Italy): Evaluation Committee
Sebastian Stücker (KIT, Germany): Evaluation Committee
François Yvon (LIMSI-CNRS, France): Program Committee

Publicity Chair

Wade Shen, (MIT, USA)
Graham Neubig, (NAIST, Japan)

Local and Financial Chair

Margit Rödder, (KIT, Germany)
Manami Matsuda, (NAIST, Japan)

Evaluation Technical Committee

Roberto Gretter (FBK, Italy): ASR Track
Sebastian Stüker (KIT, Germany): ASR Track
Mauro Cettolo (FBK, Italy): MT Track
Jan Niehues (KIT, Germany): SLT Track
Nick Ruiz, (FBK, Italy) ASR and MT Output Normalization

Language Experts

Praslav Nakov (UC Berkeley, USA) [Arabic]
Amin Farajian (FBK, Italy) [Farsi]
Shachar Mirkin (Xerox Research Centre Europe, France) [Hebrew]
Dekai Wu (HKUST, China) [Mandarin]
Krzysztof Marasek (PJIIT, Poland) [Polish]

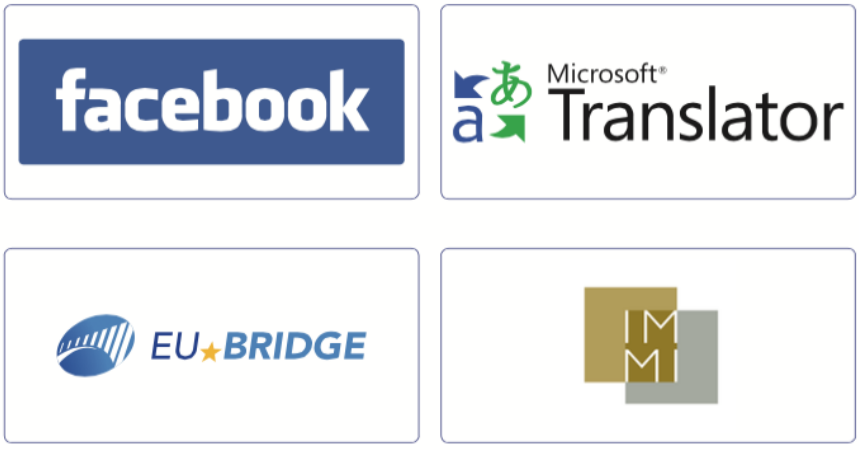
Program Committee

Alexandre Allauzen (U. Paris-Sud & LIMSI-CNRS, FR)
Gilles Adda (LIMSI, FR)
Loic Barrault (LIUM, FR)
Laurent Besacier (LIG, FR)

Alexandra Birch (Univ. of Edinburgh, UK)
Francisco Casacuberta (Univ. Politécnica de Valéncia, ES)
Mauro Cettolo (FBK, IT)
Boxing Chen (NRC, CA)
Thomas Hain (Univ. of Sheffield, UK)
Hany Hassan (Microsoft Research, US)
Xiaodong He (Microsoft Research, US)
Teresa Herrmann (KIT, DE)
Chiori Hori (NICT, JP)
Philippe Langlais (Univ. de Montréal, CA)
Yves Lepage (Waseda Univ., JP)
Qun Liu (DCU, IR)
Graham Neubig (NAIST, JP)
Jan Niehues (KIT, DE)
Michael Paul (NICT, JP)
Stephan Peitz (RWTH Aachen, DE)
Sebastian Stüker (KIT, DE)
Isabel Transcoso (INESC ID Lisboa, PT)
Hajime Tsukada (NTT, JP)
Dekai Wu (HKUST, CN)
François Yvon (LIMSI-CNRS, FR)
Joy Zhang (CMU, US)

Acknowledgments

IWSLT 2014 is proud to present its gold sponsors



Participants

	ASR			SLT		
	en	de	it	en − de	de − en	en − fr
EU-BRIDGE (EUROPE)					✔	
FBK (ITALY)	✔	✔	✔			✔
IOIT (VIETNAM)	✔					
KIT (GERMANY)	✔	✔	✔	✔	✔	✔
LIMSI (FRANCE)						✔
LIUM (FRANCE)	✔		✔			✔
MIT LL (USA)	✔		✔			
NICT (JAPAN)	✔					
RWTH (GERMANY)					✔	✔
UEDIN (UK)	✔	✔		✔	✔	✔
USFD (UK)	✔			✔		✔

Groups participating to the ASR and SLT evaluation tasks

	en − de	en − fr	en − pl	en − zh	en − ar	other directions
EU-BRIDGE (EUROPE)	⇐,⇒	⇒				
FBK (ITALY)	⇐	⇒				
HKUST (CHINA)				⇐,⇒		
KIT (GERMANY)	⇐,⇒	⇒		⇒	⇒	
LIA (FRANCE)			⇒			en⇒sl
MIRACL (TUNISIA)		⇒				
MIT LL (USA)		⇒		⇐	⇐	fa⇒en, ru⇒en
NICT (JAPAN)				⇐		
NTT-NAIST (JAPAN)	⇐,⇒					
PJWSTK (POLAND)			⇐,⇒			
RWTH (GERMANY)	⇐	⇒				
UEDIN (UK)	⇐,⇒	⇒			⇐,⇒	fa⇒en, he⇒en
USFD (UK)	⇒	⇒				
USTC (CHINA)				⇐,⇒		

Groups participating to the MT evaluation tasks

Program

Thursday, December 4th, 2014

08:30-09:15	**WORKSHOP REGISTRATION**
09:15-09:30	**Welcome Remarks** *Satoshi Nakamura (General Chair) & Alex Waibel (Steering Committee Chair)* NAIST, Japan & KIT-CMU, Germany, USA
09:30-10:30	**Report on the 11th IWSLT Evaluation Campaign, IWSLT 2014** *Mauro Cettolo, Jan Niehues, Sebastian Stücker, Luisa Bentivogli and Marcello Federico* FBK, Italy
Coffee Break (10:30-11:00)	
11:00-12:30	**EVALUATION CAMPAIGN**
11:00-11:30	**The NICT ASR System for IWSLT 2014** *Peng Shen, Xugang Lu, Xinhui Hu, Naoyuki Kanda, Masahiro Saiko and Chiori Hori* NICT, Japan
11:30-12:00	**NTT-NAIST Syntax-based SMT Systems for IWSLT 2014** *Katsuhito Sudoh, Graham Neubig, Kevin Duh and Katsuhiko Hayashi* NNT and NAIST, Japan
12:00-12:30	**LIMSI English-French Speech Translation System** *Natalia Segal, Hélène Bonneau-Maynard, Quoc Khanh Do, Alexandre Allauzen, Jean-Luc Gauvain, Lori Lamel and François Yvon* LIMSI/CNRS and Univ. Paris-Sud, France
Lunch (12:30-14:00)	
14:00-16:00	**ORAL SESSION I**
14:00-14:30	**Anticipatory Translation Model Adaptation for Bilingual Conversations** *Sanjika Hewavitharana, Dennis Mehay, Sankaranarayanan Ananthakrishnan, Rohit Kumar and John Makhoul* BBN, USA
14:30-15:00	**Better Punctuation Prediction with Hierarchical Phrase-Based Translation** *Stephan Peitz, Markus Freitag and Hermann Ney* RWTH Aachen, Germany

15:00-15:30	**Extracting Translation Pairs from Social Network Content** *Matthias Eck, Yury Zemlyanskiy, Joy Zhang and Alex Waibel* Facebook, USA
15:30-16:00	**Empirical Dependency-Based Head Finalization for Statistical Chinese-, English-, and French-to-Myanmar (Burmese) Machine Translation** *Chenchen Ding, Ye Kyaw Thu, Masao Utiyama, Andrew Finch and Eiichiro Sumita* NICT, Japan
Coffee Break (16:00-16:30)	
16:30-18:00	**POSTER SESSION I**
	Polish - English Speech Statistical Machine Translation Systems for the IWSLT 2014 *Krzysztof Wolk and Krzysztof Marasek* PJIIT, Poland
	A Topic-based Approach for Post-processing Correction of Automatic Translations *Mohamed Morchid, Stéphane Huet and Richard Dufour* LIA, France
	The NICT Translation System for IWSLT 2014 *Xiaolin Wang, Andrew Finch, Masao Utiyama, Taro Watanabe and Eiichiro Sumita* NICT, Japan
	The USTC Machine Translation System for IWSLT 2014 *Shijin Wang, Yuguang Wang, Jianfeng Li, Yiming Cui and Lirong Dai* USTC, China
	NTT-NAIST Syntax-based SMT Systems for IWSLT 2014 *Katsuhito Sudoh, Graham Neubig, Kevin Duh and Katsuhiko Hayashi* NTT and NAIST, Japan
	Improving In-Domain Data Selection For Small In-Domain Sets *Mohammed Mediani, Joshua Winebarger and Alex Waibel* KIT, Germany
	Multilingual Deep Bottle Neck Features - A Study on Language Selection and Training Techniques *Markus Müller, Sebastian Stücker, Zaid Sheikh, Florian Metze and Alex Waibel* KIT, Germany and LTI, Carnegie Mellon, USA
	Word Confidence Estimation for Speech Translation *Laurent Besacier, Benjamin Lecouteux, Luong Ngoc Quang, Kaing Hour and Marwa Hadj Salah* LIG, France
	The NAIST-NTT TED Talk Treebank *Graham Neubig, Katsuhito Sudoh, Yusuke Oda, Kevin Duh, Hajime Tsukada and Masaaki Nagata* NAIST, Japan
	Machine Translation of Multi-party Meetings: Segmentation and Disfluency Removal Strategies *Eunah Cho, Jan Niehues and Alex Waibel* KIT, Germany

	The USFD SLT System for IWSLT 2014 *Raymond W. M. Ng, Mortaza Doulaty, Rama Doddipatla, Wilker Aziz, Kashif Shah, Lucia Specia, Thomas Hain, Oscar Saz, Madina Hasan and Ghada Alharbi* Univ. Sheffield, UK
	LIMSI English-French Speech Translation System *Natalia Segal, Hélène Bonneau-Maynard, Quoc Khanh Do, Alexandre Allauzen, Jean-Luc Gauvain, Lori Lamel and François Yvon* LIMSI-CNRS and Univ. Paris-Sud, France
	The LIUM English-to-French Spoken Language Translation System and the Vecsys/LIUM Automatic Speech Recognition System for Italian Language for IWSLT 2014 *Anthony Rousseau, Loïc Barrault, Paul Deléglise, Yannick Estève, Holger Schwenk, Samir Bennacef, Armando Muscariello and Stephan Vanni* LIUM and Vecsys, France
	FBK's Machine Translation and Speech Translation Systems for the IWSLT 2014 Evaluation Campaign *Nicola Bertoldi, Prashant Mathur, Nicholas Ruiz and Marcello Federico* FBK, Italy
19:00-	**SOCIAL EVENT DINER**

Friday, December 5th, 2014

09:00-10:30	**ORAL SESSION II**
9:00-10:00	**Invited talk: Speech Translation for Everyone - Breaking down the Barriers** *Arul Menezes* Microsoft Research, USA
10:00-10:30	**Open Discussion: The future of IWSLT evaluation** *Chair: Marcello Federico and Sebastian Stücker* FBK, Italy and KIT, Germany
Coffee Break (10:30-11:00)	
11:00-12:30	**ORAL SESSION III**
11:00-11:30	**An Exploration of Segmentation Strategies in Stream Decoding** *Andrew Finch, Xiaolin Wang and Eiichiro Sumita* NICT, Japan
11:30-12:00	**Discriminative Adaptation of Continuous Space Translation Models** *Quoc-Khanh Do, Alexandre Allauzen and François Yvon* LIMSI-CNRS and Univ. Paris Sud, France
12:00-12:30	**Lexical Translation Model Using A Deep Neural Network Architecture** *Thanh-Le Ha, Jan Niehues and Alex Waibel* KIT, Germany
Lunch (12:30-14:00)	
14:00-16:00	**ORAL SESSION: QUALITY IN INTERPRETATION**
14:00-15:00	**Invited Talk: Olga Cosmidou** *Quality assurance in multilingual conference interpreting - the European Parliament experience* Directorate-General for Interpretation and Conferences, Europe
15:00-16:00	**Round Table: Quality in Interpretation** *Chair: Alex Waibel* *Participants: S. Alterberg, C. Bahr, O. Cosmidou, M. Federico, S. Stücker, D. Wu, F. Yvon*
Coffee Break (16:00-16:30)	
16:30-18:00	**POSTER SESSION II**
	Incremental Development of Statistical Machine Translation Systems *Li Gong, Aurélien Max and François Yvon* LIMSI-CNRS and Univ. Paris-Sud
	Offline Extraction of Overlapping Phrases for Hierarchical Phrase-Based Translation *Sariya Karimova, Patrick Simianer and Stefan Riezler* Heidelberg University, Germany
	Translations of the CallHome Egyptian Arabic corpus for conversational speech translation *Gaurav Kumar, Yuan Cao, Ryan Cotterell, Chris Callison-Burch, Daniel Povey and Sanjeev Khudanpur* Heidelberg University, Germany

	The MITLL-AFRL IWSLT 2014 MT System *Michaeel Kazi, Elizabeth Salesky, Brian Thompson, Jessica Ray, Michael Coury, Wade Shen, Tim Anderson, Grant Erdmann, Jeremy Gwinnup, Katherine Young, Brian Ore and Michael Hutt* MIT Lincoln Lab, USA
	The 2014 KIT IWSLT Speech-to-Text Systems for English, German and Italian *Kevin Kilgour, Michael Heck, Markus Mueller, Matthias Sperber, Sebastian Stücker and Alex Waibel* KIT, Germany
	Towards Simultaneous Interpreting: The Timing of Incremental Machine Translation and Speech Synthesis *Timo Baumann, Srinivas Bangalore and Julia Hirschberg* Univ. Hamburg, Germany, ATT Labs and Columbia Univ., USA
18:00-	**CLOSING REMARKS + ANNOUNCEMENTS**

Keynotes

Speech translation for everyone - breaking down the barriers

Arul Menezes, Microsoft Research

Abstract

Fifty years ago Star Trek had the Universal Translator. Thirty-five years ago we were introduced to the babel fish in The Hitchhiker's Guide to the Galaxy. Decades later, is reality finally catching up to science fiction? Given the enormous strides made in speech recognition and machine translation over the last decade, is this just as matter of chaining speech recognition and machine translation together?

In the Skype Translator project we set ourselves an ambitious goal - to enable successful open-domain conversations between Skype users in different parts of the world, speaking different languages. As one might imagine, putting together two error-prone technologies such as speech recognition and machine translation raises some unique challenges.

In this talk, I will share what we have learned over the course of the Skype Translator project. I will discuss what we are doing to bridge the gap between ASR and MT, how we are adapting our ASR and MT systems to the real world challenges presented by our open-domain conversational scenario, and what it takes to get this technology into the hands of real users. I will also touch upon some of the open issues and challenges we still face.

Bio

Arul Menezes heads the Machine Translation team at Microsoft Research. Over the past 15 years, he has driven Machine Translation at Microsoft Research from a basic research project to a web-scale production service with a variety of offerings for consumers and businesses, and millions of users worldwide. These include the Bing Translator and the Microsoft Translator Hub customization service, as well as the upcoming Skype Translator product. The Microsoft MT system is based on the treelet translation approach to syntactic statistical MT, co-invented by Arul, Chris Quirk and Colin Cherry. The MSR MT team integrates research and product development in a single team, covering everything from MT modelling and algorithms to data gathering and delivery of the live web service. This eliminates the traditional "tech transfer" from research to product, and enables the team to get research breakthroughs into customer hands without delay. Arul was educated at the Indian Institute of Technology, Bombay and at Stanford University.

Evaluation Campaign

Report on the 11th IWSLT Evaluation Campaign, IWSLT 2014

Mauro Cettolo[(1)] Jan Niehues[(2)] Sebastian Stüker[(2)] Luisa Bentivogli[(1)] Marcello Federico[(1)]

[(1)] FBK - Via Sommarive 18, 38123 Trento, Italy
[(2)] KIT - Adenauerring 2, 76131 Karlsruhe, Germany

Abstract

The paper overviews the 11th evaluation campaign organized by the IWSLT workshop. The 2014 evaluation offered multiple tracks on lecture transcription and translation based on the TED Talks corpus. In particular, this year IWSLT included three automatic speech recognition tracks, on English, German and Italian, five speech translation tracks, from English to French, English to German, German to English, English to Italian, and Italian to English, and five text translation track, also from English to French, English to German, German to English, English to Italian, and Italian to English. In addition to the official tracks, speech and text translation optional tracks were offered, globally involving 12 other languages: Arabic, Spanish, Portuguese (B), Hebrew, Chinese, Polish, Persian, Slovenian, Turkish, Dutch, Romanian, Russian. Overall, 21 teams participated in the evaluation, for a total of 76 primary runs submitted. Participants were also asked to submit runs on the 2013 test set (progress test set), in order to measure the progress of systems with respect to the previous year. All runs were evaluated with objective metrics, and submissions for two of the official text translation tracks were also evaluated with human post-editing.

1. Introduction

This paper overviews the results of the 2014 evaluation campaign organized by the International Workshop of Spoken Language Translation. The IWSLT evaluation has been running now for over a decade and has offered along these years a variety of speech translation tasks [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. The 2014 IWSLT evaluation continued along the line set in 2010, by focusing on the translation of TED Talks, a collection of public speeches covering many different topics. As in the previous two years, the evaluation included tracks for all the core technologies involved in the spoken language translation task, namely:

- Automatic speech recognition (ASR), i.e. the conversion of a speech signal into a transcript,
- Spoken language translation (SLT), that addressed the conversion and translation of a speech signal into a transcript in another language,
- Machine translation (MT), i.e. the translation of a polished transcript into another language.

However, with respect to previous rounds, new languages have been added to each track. The ASR track that previously included German and English, was extended by Italian. The SLT and MT track offered official English-French, English-German, German-English, English-Italian, and Italian-English translation directions. Besides the official evaluation tracks, many other optional translation directions were also offered. Optional SLT directions were English-Arabic and English-Chinese. Optional MT translation directions were: English from/to Arabic, Spanish, Portuguese (B), Hebrew, Chinese, Polish, Persian, Slovenian, Turkish, Dutch, Romanian, and Russian. For each official and optional translation direction, training and development data were supplied by the organizers through the workshop's website. Major parallel collections made available to the participants were the WIT^3 [11] corpus of TED talks, all data from the WMT 2014 workshop [12], the MULTIUN corpus, and the SETimes parallel corpus. A list of monolingual resources was provided too, that includes both freely available corpora and corpora available from LDC. Test data were released at the beginning of each test period, requiring participants to return one primary run and optional contrastive runs within one week. The schedule of the evaluation was organized as follows: June 2, release of training data; Sept 1–10, ASR test period; Sept 16–25, SLT test period (official directions); Sept 26–Oct 5, MT test period (official directions); Oct 6–17, MT and SLT test period of all optional directions.

All runs submitted by participants were evaluated with automatic metrics. In addition, manual evaluation was carried out for two MT tracks, namely the English-French and English-German tracks. Following the methodology introduced last year, systems were evaluated by calculating HTER values on post-edits created by professional translators. The rational behind this evaluation is to assess the utility of an MT output by measuring the post-editing effort needed by a professional translator to fix it.

This year, 21 sites participated (see Table 1) submitting a total of 76 primary runs: 15 to the ASR track, 16 to the SLT track, and 45 to the MT track (see Sections 3.3, 4.3, 5.3 for details).

In the rest of the paper we first outline the main goals of the IWSLT evaluation and then each single track in detail, in particular: its specifications, supplied language resources, evaluation methods, and results. The paper ends with some concluding remarks about the experiences gained in this eval-

Proceedings of the 11th International Workshop on Spoken Language Translation
Lake Tahoe, December 4th and 5th, 2014

uation exercise, followed by appendixes that complement the information given in the specific sections.

2. TED Talks

2.1. TED events

The translation of TED talks was introduced for the first time at IWSLT 2010. TED is a nonprofit organization that "invites the world's most fascinating thinkers and doers [...] to give the talk of their lives". Its website[1] makes the video recordings of the best TED talks available under the Creative Commons license. All talks have English captions, which have also been translated into many languages by volunteers worldwide. In addition to the official TED events held in North America, a series of independent TEDx events are regularly held around the world, which share the same format of the original TED talks but are hold in the language of the hosting country. Recently, an effort was made to set up a web repository [11] that distributes dumps of the available TED talks transcripts and translations under form of parallel texts, ready to use for training and evaluating MT systems.

Besides representing a popular benchmark for spoken language technology, the TED Talks task embeds interesting research challenges which are unique among the available speech recognition and machine translation benchmarks. TED Talks is a collection of rather short speeches (max 18 minutes each, roughly equivalent to 2,500 words) which cover a wide variety of topics. Each talk is delivered in a brilliant and original style by a very skilled speaker and, while addressing a wide audience, it pursues the goal of both entertaining and persuading the listeners on a specific idea. From the point of view of ASR, TED talks require copying with background noise – e.g. applauses and laughs by the public –, different accents including non native speakers, varying speaking rates, prosodic aspects, and, finally, narrow topics and personal language styles. From an application perspective, TED Talks transcription is the typical life captioning scenario, which requires producing polished subtitles in real-time.

From the point of view of machine translation, translating TED Talks implies dealing with spoken rather than written language, which is hence expected to be structurally less complex, formal and fluent. Moreover, as human translations of the talks are required to follow the structure and rythm of the English captions,[2] a lower amount of rephrasing and reordering is expected than in ordinary translation of written documents.

From an application perspective, TED Talks suggest translation tasks ranging from off-line translation of written captions, up to on-line speech translation, requiring a tight integration of MT with ASR possibly handling stream-based processing.

[1] http://www.ted.com

[2] See recommendations to translators in http://translations.ted.org/wiki.

3. ASR Track

3.1. Definition

The goal of the *Automatic Speech Recognition* (ASR) track for IWSLT 2014 was to transcribe English TED talks, as well as German and Italian TEDx talks. The speech in TED lectures is in general planned, well articulated, and recorded in high quality. The main challenges for ASR in these talks are to cope with a large variability of topics, the presence of non-native speakers, and the rather informal speaking style. For the TEDx talks the recording conditions are a little bit more difficult than for the English TED talks. While the TEDx talks aim to mimic the TED talks, they are not as well prepared and well rehearsed as the TED lectures, and recording is often done by amateurs resulting in often poorer recording quality than for the TED lectures.

The result of the recognition of the talks is used for two purposes. It is used to measure the performance of ASR systems on the talks and it is used as input for the spoken language translation evaluation (SLT), see Section 4.

3.2. Evaluation

Participants had to submit the results of the recognition of the tst2014 set in CTM format. The word error rate was measured case-insensitive. After the end of the evaluation a preliminary scoring was performed with the first set of references. This was followed by an adjudication phase in which participants could point out errors in the reference transcripts. The adjudication results were collected and combined into the final set of references with which the official scores were calculated.

In order to measure the progress of the systems over the years on English and German, participants also had to provide results on the test set from 2013, i.e. *tst2013*.

3.3. Submissions

For this year's evaluation we received primary submissions from eight sites as well as one combined submission by the EU-BRIDGE project. Seven sites participated in the English evaluation, three sites in the German evaluation and four sites in the Italian one. For English we further received a total of seven contrastive submissions from five sites. For German we received three contrastive submissions from one participant. For Italian we receieved five contrastive submissions from three sites. Also, for English we received a joint submission by the project EU-BRIDGE which was a ROVER combination of the partners' outputs and for which no separate system description was submitted.

3.4. Results

The detailed results of the primary submissions of the evaluation in terms of word error rate (WER) can be found in Appendix A.1. The word error rate of the submitted systems in in the range of 8.4%–19.7% for English, 24.0%–38.8% for

Table 1: List of Participants

EU-BRIDGE	RWTH& UEDIN& KIT& FBK[13]
FBK	Fondazione Bruno Kessler, Italy [14, 15]
HKUST	Hong Kong University of Science and Technology, Hong Kong [16]
IOIT	Inst. of Inform. and Techn., Vietn. Acad. of Science and Techn. & Thai Nguyen University, Vietnam[17]
KIT	Karlsruhe Institute of Technology, Germany [18, 19]
KLE	Pohang University of Science and Technology, Republic of Korea
LIA	Laboratoire Informatique d'Avignon (LIA) University of Avignon, France [20]
LIMSI	LIMSI - LIMSI, France [21]
LIUM	LIUM, University of Le Mans, France [22]
MIRACL	MIRACL Laboratory Pôle Technologique, Tunisia & LORIA Nancy, France [23]
MITLL-AFRL	Mass. Institute of Technology/Air Force Research Lab., USA
NICT	National Institute of Communications Technology, Japan [24, 25]
NTT-NAIST	NTT Communication Science Labs, Japan & NAIST[26]
PJIIT	Polish-Japanese Institute of Information Technology, Poland [27]
RWTH	Rheinisch-Westfälische Technische Hochschule Aachen, Germany [28]
SFAX	Sfax University, Tunisia
UEDIN	University of Edinburgh, United Kingdom [29, 30]
UMONTREAL	Université de Montréal, Canada
USFD	University of Sheffield, United Kingdom [31]
USTC	National Engineering Laboratory of Speech and Lang. Inform. Proc., Univ. of Science and Techn. of China [32]
VECSYS-LIUM	Vecsys Technologies, France & University of Le Mans, France [22]

German, and 21.9%–25.4% for Italian.

In German, the fact that TEDx have sometimes worse recording conditions than TED talks was reflected by the fact that two talks in the German tst2014 had WERs above 40%. WERs for all other talks were in the range from 9% to 32%.

For English, it can be seen that all participants from IWSLT 2013 made progress, many significant progress, e.g., bringing down the WER from 13.5% to 10.6% on *tst2013*, a relative reduction of 21% over the course of one year. For German, the best performing system only made minor progress, while one of the runner-ups made significant progress and one participant essentially stood the same.

4. SLT Track

4.1. Definition

The SLT track required participants to translate the English, German and Italian talks of *tst2014* from the audio signal (see Section 3). The challenge of this translation task over the MT track is the necessity to deal with automatic, and in general error prone, transcriptions of the audio signal, instead of correct human transcriptions.

For German and Italian, participants had to translate into English. For English as source language, participants had to translate into French. In addition, participants could also optionally translate from English into one of the following languages: German, Italian, Arabic and Mandarin Chinese.

4.2. Evaluation

For the evaluation, participants could choose to either use their own ASR technology, or to use ASR output provided by the conference organizers. In order to facilitate scoring, participants had to segment the audio according to the manual reference segmentation provided by the organizers of the evaluation.

For English, the ASR output provided by the organizers was a ROVER combination of the output from five submissions to the ASR track. The result of the ROVER had a WER of 8.2%. For German and Italian we used the two single best scored submissions, as ROVER combination with other systems did not give any performance gains.

The results of the translation had to be submitted in the same format as for the machine translation track (see Section 5).

4.3. Submissions

We received 16 primary and 31 contrastive submissions from nine participants, English to French receiving the most submissions.

4.4. Results

The detailed results of the automatic evaluation in terms of BLEU and TER can be found in Appendix A.1.

Table 2: Monolingual resources for official language pairs

data set	lang	sent	token	voc
train	De	183k	3.36M	124.7k
	En	188k	3.81M	63.4k
	Fr	186k	4.00M	77.0k
	It	185k	3.49M	90.2k

5. MT Track

5.1. Definition

The MT TED track basically corresponds to a subtitling translation task. The natural translation unit considered by the human translators volunteering for TED is indeed the single caption — as defined by the original transcript — which in general does not correspond to a sentence, but to fragments of it that fit the caption space. While translators can look at the context of the single captions, arranging the MT task in this way would make it particularly difficult, especially when word re-ordering across consecutive captions occurs. For this reason, we preprocessed all the parallel texts to re-build the original sentences, thus simplifying the MT task.

For each official and optional translation direction, in-domain training and development data were supplied through the website of WIT[3] [11], while out-of-domain training data through the workshop's website. As usual, some of the talks added to the TED repository during the last year have been used to define the new evaluation sets (*tst2014*), while the remaining new talks have been included in the training sets. For reliably assessing progress of MT systems over the years, the evaluation sets *tst2013* of edition 2013 were distributed together with *tst2014* as progressive test sets, when available. Development sets (*dev2010, tst2010, tst2011* and *tst2012*) are either the same of past editions or, in case of new language pairs, have been built upon the same talks.

Evaluation sets *tst2014* of $DeEn$ and $ItEn$ MT tasks derive from those prepared for ASR/SLT tracks, which consist of TEDx talks delivered in German and Italian language, respectively; therefore, no overlap exists with any other TED talk involved in other tasks. Since the $DeEn$ TEDx based MT task was proposed in 2013 as well, the *tst2013* has been released as progressive test set; on the contrary, it is the first time that Italian is involved in ASR/SLT tracks, therefore no evaluation set is available for assessing progress. A single TEDx based development set was released for each pair, together with standard TED based development sets *dev2010, tst2010, tst2011* and *tst2012* sets.

Tables 2 and 3 provides statistics on in-domain texts supplied for training, development and evaluation purposes for the official directions.

MT baselines were trained from TED data only, i.e. no additional out-of-domain resources were used. The standard tokenization via the tokenizer script released with the Europarl corpus [33] was applied to all languages, with the exception of Chinese and Arabic languages, which were preprocessed by, respectively: the Stanford Chinese Segmenter [34] and the QCRI-normalizer.[3]

Table 3: Bilingual resources for official language pairs.

MT task	set	sent	tokens		talks
En→Fr			En	Fr	
	train	179k	3.63M	3.88M	1415
	TED.dev2010	887	20,1k	20,2k	8
	TED.tst2010	1,664	32,0k	33,9k	11
	TED.tst2011	818	14,5k	15,6k	8
	TED.tst2012	1,124	21,5k	23,5k	11
	TED.tst2013	1,026	21,7k	23,3k	16
	TED.tst2014	1,305	24,8k	27,5k	15
En↔De			En	De	
	train	172k	3.46M	3.24M	1361
	TED.dev2010	887	20,1k	19,1k	8
	TED.tst2010	1,565	32,0k	30,3k	11
	TED.tst2011	1,433	26,9k	26,3k	16
	TED.tst2012	1,700	30,7k	29,2k	15
→	TED.tst2013	993	20,9k	19,7k	16
	TED.tst2014	1,305	24,8k	23,8k	15
←	TEDx.dev2012	1,165	21,6k	20,8k	7
	TEDx.tst2013	1,363	23,3k	22,4k	9
	TEDx.tst2014	1,414	28,1k	27,6k	10
En↔It			En	It	
	train	182k	3.68M	3.44M	1434
	TED.dev2010	887	20,1k	17,9k	8
	TED.tst2010	1,529	31,0k	28,7k	10
	TED.tst2011	1,433	26,9k	24,5k	16
	TED.tst2012	1,704	30,7k	28,2k	15
→	TED.tst2013	1,402	30,1k	28,7k	21
	TED.tst2014	1,183	22,6k	21,2k	14
←	TEDx.dev2014	1,056	28,9k	28,6k	13
	TEDx.tst2014	883	25,9k	26,5k	13

The baselines were developed with the Moses toolkit. Translation and lexicalized reordering models were trained on the parallel training data; 5-gram LMs with improved Kneser-Ney smoothing were estimated on the target side of the training parallel data with the IRSTLM toolkit. The weights of the log-linear interpolation model were optimized with the MERT procedure provided with Moses, mostly on the development sets *tst2010*; the exceptions are: TEDx tasks, where the TEDx based development sets were used; the two pairs involving Slovenian, where *dev2012* were employed.

5.2. Evaluation

The participants to the MT track had to provide the results of the translation of the test sets in NIST XML format. The output had to be case-sensitive and had to contain punctuation

[3] QCRI-normalizer was specifically developed for IWSLT Evaluation Campaigns by P. Nakov and F. Al-Obaidli at Qatar Computing Research Institute.

(case+punc).

The quality of the translations was measured automatically against the human translations created by the TED open translation project, and by human subjective evaluation (Section 5.5). Tokenization scripts were applied automatically to all run submissions prior to evaluation.

Evaluation scores were calculated for the two automatic standard metrics BLEU and TER, as implemented in mteval-v13a.pl[4] and tercom-0.7.25[5], respectively.

5.3. Submissions

We received submissions from 14 different sites. On official pairs, the total number of primary runs is 39: 20 on *tst2014* and 19 on *tst2013*; 15 primary runs regard the $EnFr$ pair, 10 the $EnDe$ and 14 the $DeEn$; in addition, we were asked to evaluate also 64 contrastive runs.

Concerning the optional pairs, we received 48 primary runs (25 on *tst2014* and 23 on *tst2013*) and 20 contrastive submissions. The tasks that attracted the most interest are those involving Chinese: 8 primary runs were submitted for $EnZh$, 8 for $ZhEn$. The other submissions involve Arabic, Polish, Farsi, Hebrew, Turkish and Slovenian.

5.4. Results

Table 4: BLEU and TER scores of baseline SMT systems on all *tst2014* sets. (†) TEDx test set. (*) Char-level scores.

pair		direction → BLEU	direction → TER	direction ← BLEU	direction ← TER
En	Fr	32.07	48.62	–	–
	De	18.33	62.11	†17.89	†64.91
	It	27.15	53.19	†26.12	†55.30
	Ar	11.13	73.01	20.59	62.62
	Es	31.31	48.29	33.88	45.96
	Fa	11.31	71.20	16.74	72.02
	He	15.91	65.62	24.41	58.38
	Nl	22.77	58.38	27.82	52.98
	Pl	9.63	82.81	14.28	68.96
	Pt	31.25	47.25	36.44	42.80
	Ro	18.05	65.25	25.06	54.62
	Ru	11.74	71.99	15.91	69.73
	Sl	8.46	73.94	14.27	71.03
	Tr	7.75	78.69	12.88	77.15
	Zh	*16.49	*79.50	11.74	72.31

First of all, for reference purposes Table 4 shows BLEU and TER scores on the *tst2014* evaluation sets of the baseline systems we developed as described in Section 5.1.

The results on the official test set for each participant are shown in Appendix A.1. For most languages, we show the case-sensitive and case-insensitive BLEU and TER scores. In contrast to the other language pairs, for English to Chinese character-level scores are reported.

These results also show again the scores of the baseline system. Thereby, it is possible to see the improvements of the submitted systems on the different languages over the baseline system.

In Appendix A.2 the results on the progress test sets *test2013* are shown. When comparing the results to the submissions from last year, the performance could be improved in nearly all tasks.

5.5. Human Evaluation

Human evaluation was carried out on primary runs submitted by participants to two of the official MT TED tracks, namely the MT English-German ($EnDe$) track and MT English-French ($EnFr$) track. Following the methodology introduced last year, human evaluation was based on *Post-Editing*, and HTER (Human-mediated Translation Edit Rate) was adopted as the official evaluation metric to rank the systems.

Post-Editing, i.e. the manual correction of machine translation output, has long been investigated by the translation industry as a form of machine assistance to reduce the costs of human translation. Nowadays, Computer-aided translation (CAT) tools incorporate post-editing functionalities, and a number of studies [35, 36] demonstrate the usefulness of MT to increase professional translators' productivity. The MT TED task offered in IWSLT can be seen as an interesting application scenario to test the utility of MT systems in a real subtitling task.

From the point of view of the evaluation campaign, our goal was to adopt a human evaluation framework able to maximize the benefit to the research community, both in terms of information about MT systems and data and resources to be reused. With respect to other types of human assessment, such as judgments of translation quality (i.e. adequacy/fluency and ranking tasks), the post-editing task has the double advantage of producing *(i)* a set of edits pointing to specific translation errors, and *(ii)* a set of additional reference translations. Both these byproducts are very useful for MT system development and evaluation. Furthermore, HTER[37] - which consists of measuring the minimum edit distance between the machine translation and its manually post-edited version - has been shown to correlate quite well with human judgments of MT quality.

The human evaluation setup and the collection of post-editing data are presented in Section 5.5.1, whereas the results of the evaluation are presented in Section 5.5.2.

5.5.1. Evaluation Setup and Data Collection

The human evaluation (HE) dataset created for each MT track was a subset of the corresponding 2013 progress test

[4] http://www.itl.nist.gov/iad/mig/tests/mt/2009/

[5] http://www.cs.umd.edu/ snover/tercom/

set *(tst2013)*.[6] Both the $EnDe$ and $EnFr$ *tst2013* datasets are composed of 16 TED Talks, and we selected around the initial 60% of each talk. This choice of selecting a consecutive block of sentences for each talk was determined by the need of realistically simulating a caption post-editing task on several TED talks. The resulting HE sets are composed of 628 segments for $EnDe$ and 622 segments for $EnFr$, both corresponding to around 11,000 words.

In order to evaluate the MT systems, the *bilingual* post-editing task was chosen, where professional translators are required to post-edit the MT output directly according to the source sentence. Bilingual post-editing is expected to give more accurate results than monolingual post-editing as post-editors do not depend on an given - and possibly imprecise - translation. Then, HTER scores were calculated on the created post-edits. HTER [37] is a semi-automatic metric derived from TER (Translation Edit Rate). TER measures the amount of editing that a human would have to perform to change a machine translation so that it exactly matches a given reference translation. HTER is a variant of TER where a new reference translation is generated by applying the minimum number of post-edits to the given MT output. This new *targeted* reference is then used as the only reference translation to calculate the TER of the MT output.

An interesting outcome of last year's manual evaluation [10] was that the most informative and reliable HTER was not obtained by using only the targeted reference but by exploiting all the post-edits of the evaluated MT outputs. According to these results, also this year systems were officially ranked according to HTER calculated on multiple references.

As for the systems to be evaluated, this year we received five primary runs for the $EnDe$ track and seven for the $EnFr$ track. All the five $EnDe$ MT outputs were post-edited, whereas for the $EnFr$ track we decided to post-edit only five MT outputs out of the seven received. This reduction is not supposed to affect the official evaluation results - since all the participating systems are evaluated with HTER based on multiple post-edits - and it allowed us to respect the budget limitations while offering the community five additional reference translations for a high number of segments (around 60% of the test sets) and for two different language pairs. The five MT outputs selected for post-editing in the $EnFr$ task are the top-5 ranked systems according to automatic evaluation (see Appendix A).

In the preparation of the post-editing data to be collected, some constraints were identified to ensure the soundness of the evaluation: *(i)* each translator must post-edit all segments of the HE set, *(ii)* each translator must post-edit the segments of the HE set only once, and *(iii)* each MT system must be equally post-edited by all translators. Furthermore, in order to cope with the variability of post-editors (i.e. some translators could systematically post-edit more than others) we devised a scheme that dispatches MT outputs to translators both randomly and satisfying the uniform assignment constraints. For each task, five documents were hence prepared including all source segments of the HE set and, for each source segment, one MT output selected from one of the five systems.

Documents were delivered to a language service provider together with instructions to be passed on to the translators, and the post-editing tasks were run using an enterprise-level CAT tool developed under the MateCat project[7]. Both the post-editing interface and the guidelines given to translators are presented in Appendix B.

For each task, the resulting collected data consist of five new reference translations for each of the sentences of the HE set. Each one of these five references represents the targeted translation of the system output from which it was derived. From the point of view of the system output, one targeted translation and other four translations are available.

The main characteristics of the work carried out by post-editors are presented in Table 5 for the $EnDe$ task and in Table 6 for the $EnFr$ task, and largely confirm last year's findings. In the tables, the post-editing effort for each translator is given. Post-editing effort is to be interpreted as the number of actual edit operations performed to produce the post-edited version and - consequently - it is calculated as the HTER of all the system sentences post-edited by each single translator. It is interesting to see that the PE effort is similar for both language pairs, and also highly variable among post-editors, ranging from 19.69% to 40.91% for the $EnDe$ task, and from 17.47% to 39.65% for the $EnFr$ task. Data about weighted standard deviation confirm post-editor variability, showing that the five translators produced quite different post-editing effort distributions.

Table 5: En-De task: Post-editing information for each Post-editor

PEditor	PE Effort	*std-dev*	Sys TER	*std-dev*
PE 1	32.17	18.80	56.05	20.23
PE 2	19.69	13.56	56.32	20.34
PE 3	40.91	17.23	56.18	19.58
PE 4	27.56	14.71	55.93	20.02
PE 5	24.99	15.62	55.63	19.88

Table 6: En-Fr task: Post-editing information for each Post-editor

PEditor	PE Effort	*std-dev*	Sys TER	*std-dev*
PE 1	34.96	20.21	42.60	17.61
PE 2	17.47	14.76	42.81	17.98
PE 3	23.68	14.17	43.02	17.74
PE 4	39.65	20.47	42.27	17.78
PE 5	19.73	14.07	42.86	17.72

[6] Since all the data produced for human evaluation will be made publicly available thorough the WIT^3 repository, we used the 2013 test set in order to keep the 2014 test set blind to be used as a progress test for next year's evaluation.

[7] www.matecat.com

To further study post-editor variability, we exploited the official reference translations available for the two TED tracks and we calculated the TER of the MT outputs assigned to each translator for post-editing ("Sys TER" Column in Tables 5 and 6), as well as the related standard deviation.

As we can see from the tables, the documents presented to translators (composed of segments produced by different systems) are very homogeneous, as they show very similar TER scores and standard deviation figures. This also confirms that the procedure followed in data preparation was effective.

The variability observed in post-editing effort - despite the similarity of the input documents - is most probably due to translators' subjectivity in carrying out the post-editing task. Thus, post-editor variability is an issue to be addressed to ensure a sound evaluation of the systems.

5.5.2. *Evaluation Results*

As anticipated above, last year's human evaluation results demonstrated that HTER computed against all the references produced by all post-editors allowed a more reliable and consistent evaluation of MT systems with respect to HTER calculated against the targeted reference only. Indeed, the HTER reduction obtained using all post-edits clearly showed that exploiting all the available reference translations is a viable way to control and overcome post-editors' variability. For this reason, also this year systems were officially ranked according to HTER calculated on multiple references.

For the $EnDe$ task, HTER was calculated using all the five post-edits available, i.e. for each system the targeted translation and the additional four references were used. For the $EnFr$ task, since the post-edits for two MT outputs had not been created, in order to avoid biases only four post-edits out of five were used to calculate HTER, namely excluding from each system's evaluation its targeted translation.

The official results of human evaluation are given in Tables 7 and 8, which also present a comparison of HTER scores and rankings with TER results - on the HE set and on the full test set - calculated against the official reference translation used for automatic evaluation (see Section 5.2).[8] For the $EnFr$ task, the official HTER results presented in Table 8 for FBK and MIRACL (which do not have a corresponding post-edit) are those obtained on the combination of the four post-edits which gave the best results.

In general, the very low HTER results obtained in both tasks demonstrate that the overall quality of the systems is very high. Moreover, all systems are very close to each other. To establish the reliability of system ranking, for all pairs of systems we calculated the statistical significance of the observed differences in performance. Statistical significance was assessed with the *approximate randomization* method [38], a statistical test well-established in the NLP community [39] and that, especially for the purpose of MT evaluation, has been shown [40] to be less prone to type-I errors than the bootstrap method [41]. The approximate randomization test was based on 10,000 iterations, and differences were considered statistically significant at $p < 0.01$. According to this test, for both tasks a winning system cannot be indicated, as there is no system that is significantly better than all other systems. In particular, for the $EnDe$ task only the bottom-ranking system (KLE) is significantly worse than all the other systems. For the $EnFr$ task, in Table 8 we report - next to the HTER score of each system - the name of the first system in the ranking with respect to which differences are statistically significant. We can see that only the two top-ranking systems are significantly better than the four bottom-ranking systems (from UEDIN to MIRACL), whereas all the other systems significantly differ only with respect to MIRACL.

Furthermore, for comparison purposes, Table 8 presents additional HTER results calculated on all the five post-edits available for the $EnFr$ task. First, it is interesting to note the further HTER reduction achieved, especially for the five top-scoring systems since their corresponding targeted reference was added. Also, comparing the two language pairs, we see that the HTER scores obtained for $EnFr$ with five reference translations are overall lower than those obtained for $EnDe$, indicating that systems translating into French perform better than systems translating into German.

A number of additional observations can be drawn by comparing the official HTER results with TER results. In general, for both tasks we can see that HTER reduces the edit rate of more than 50% with respect to TER. Moreover,

Table 7: En-De Task: Official human evaluation results

System Ranking	**HTER** ***HE Set 5 PErefs***	**TER** HE Set ref	**TER** Test Set ref
EU-BRIDGE	**19.22**	54.55	53.62
UEDIN	**19.93**	56.32	55.12
KIT	**20.88**	54.88	53.83
NTT-NAIST	**21.32**	54.68	53.86
KLE	**28.75**	59.67	58.27
Rank Corr.		0.60	0.70

Table 8: En-Fr Task: Official human evaluation results

System Ranking	**HTER** ***HE Set 4 PErefs***	**HTER** HE Set 5 PErefs	**TER** HE Set ref	**TER** Test Set ref
EU-BRIDGE	**19.21**$^{\text{UEDIN}}$	16.48	42.64	43.27
RWTH	**19.27**$^{\text{UEDIN}}$	16.55	41.82	42.58
KIT	**20.89**$^{\text{MIRACL}}$	17.64	42.33	43.09
UEDIN	**21.52**$^{\text{MIRACL}}$	17.23	43.28	43.80
MITLL-AFRL	**22.64**$^{\text{MIRACL}}$	18.69	43.48	44.05
FBK	**22.90**$^{\text{MIRACL}}$	22.29	44.28	44.83
MIRACL	**33.61**	32.90	52.19	51.96
Rank Corr.		0.96	0.90	0.90

[8] Note that since HTER and TER are edit-distance measures, lower numbers indicate better performances

the correlation between evaluation metrics is measured using *Spearman's rank correlation coefficient* $\rho \in$ [-1.0, 1.0], with ρ = 1.0 if all systems are ranked in same order, ρ = -1.0 if all systems ranked in reverse order and ρ = 0.0 if no correlation exists. We can see from the tables that TER rankings correlate well with the official HTER.

To conclude, the post-editing task introduced this year for manual evaluation brought benefit to the IWSLT community, and in general to the MT field. In fact, producing post-edited versions of the participating systems' outputs allowed us to carry out a quite informative evaluation by minimizing the variability of post-editors, who naturally tend to diverge from the post-editing guidelines and personalize their translations. Moreover, a number of additional reference translations will be available for further development and evaluation of MT systems.

6. Conclusions

We have reported on the evaluation campaign organized for the eleventh edition of the IWSLT workshop. The evaluation has addressed three tracks: automatic speech recognition of talks (in English, German, and Italian), speech-to-text translation, and text-to-text translation, both from German to English, English to German, and English to French. Besides the official translation directions, many optional translation tasks were available, too, including 12 additional languages. For each task, systems had to submit runs on three different test sets: a newly created official test set, and a progress test set created and used for the 2013 evaluation. This year, 21 participants took part in the evaluation, submitting a total of 76 primary runs, which were all scored with automatic metrics. We also manually evaluated runs of the English-German and English-French text translation tracks. In particular, we asked professional translators to post-edit system outputs on a subset of the 2013 progress test set, in order to produce *close references* for them. While we have observed a significant variability among translators, in terms of post-edit effort, we could obtain more reliable scores by using all the produced post-edits as reference translations. By using the HTER metric, for both tracks the post-edit effort of the best performing system results remarkably low, namely around 19%. Considering that this is still an upper bound of the ideal HTER score, this percentage of post-editing seems to be another strong argument supporting the utility of machine translation for human translators.

7. Acknowledgements

Research Group 3-01' received financial support by the '*Concept for the Future*' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative. The work leading to these results has received funding from the European Union under grant agreement no 287658 — Bridges Across the Language Divide (EU-BRIDGE).

8. References

[1] Y. Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, and J. Tsujii, "Overview of the IWSLT04 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2004, pp. 1–12.

[2] M. Eck and C. Hori, "Overview of the IWSLT 2005 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Pittsburgh, PA, 2005, pp. 1–22.

[3] P. Michael, "Overview of the IWSLT 2006 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2006, pp. 1–15.

[4] C. S. Fordyce, "Overview of the IWSLT 2007 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Trento, Italy, 2007, pp. 1–12.

[5] M. Paul, "Overview of the IWSLT 2008 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Waikiki, Hawaii, 2008, pp. 1–17.

[6] ——, "Overview of the IWSLT 2009 Evaluation Campaign," in *Proceedings of the sixth International Workshop on Spoken Language Translation*, Tokyo, Japan, 2009, pp. 1–18.

[7] M. Paul, M. Federico, and S. Stüker, "Overview of the IWSLT 2010 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Paris, France, 2010, pp. 3–27.

[8] M. Federico, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2011 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, San Francisco, USA, 2011, pp. 11–27.

[9] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Hong Kong, HK, 2012, pp. 11–27.

[10] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 10th IWSLT Evaluation Campaign," in *Proceedings of the Tenth International Workshop on Spoken Language Translation (IWSLT 2013)*, Heidelberg, Germany, 2013.

[11] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of the Annual Conference of the European Association for Machine Translation*

(EAMT), Trento, Italy, May 2012. [Online]. Available: http://hltshare.fbk.eu/EAMT2012/html/Papers/59.pdf

[12] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna, "Findings of the 2014 Workshop on Statistical Machine Translation," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, MD, USA, 2014.

[13] M. Freitag, J. Wuebker, S. Peitz, H. Ney, M. Huck, A. Birch, N. Durrani, P. Koehn, M. Mediani, I. Slawik, J. Niehues, E. Cho, A. Waibel, N. Bertoldi, M. Cettolo, and M. Federico, "Combined Spoken Language Translation," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[14] B. Babaali, R. Serizel, S. Jalalvand, D. Falavigna, R. Gretter, and D. Giuliani, "FBK @ IWSLT 2014 - ASR track," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[15] N. Bertoldi, P. Mathur, N. Ruiz, and M. Federico, "FBK's Machine Translation and Speech Translation Systems for the IWSLT 2014 Evaluation Campaign," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[16] M. Beloucif, C.-K. Lo, and D. Wu, "Improving tuning against MEANT," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[17] Q. B. Nguyen, T. T. Vu, and C. M. Luong, "The Speech Recognition Systems of IOIT for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[18] K. Kilgour, M. Heck, M. Müller, M. Sperber, S. Stüker, and A. Waibel, "The 2014 KIT IWSLT Speech-to-Text Systems for English, German and Italian," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[19] I. Slawik, M. Mediani, J. Niehues, Y. Zhang, E. Cho, T. Herrmann, T.-L. Ha, and A. Waibel, "The KIT Translation Systems for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[20] M. Morchid, S. Huet, and R. Dufour, "A Topic-based Approach for Post-processing Correction of Automatic Translations," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[21] N. Segal, H. Bonneau-Maynard, Q. K. Do, A. Allauzen, J.-L. Gauvain, L. Lamel, and F. Yvon, "LIMSI English-French Speech Translation System," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[22] A. Rousseau, L. Barrault, P. Deléglise, Y. Estève, H. Schwenk, S. Bennacef, A. Muscariello, and S. Vanni, "The LIUM English-to-French Spoken Language Translation System and the Vecsys/LIUM Automatic Speech Recognition System for Italian Language for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[23] A. B. Romdhane, S. Jamoussi, A. B. Hamadou, and K. Smaili, "Phrase-based Language Modelling for Statistical Machine Translation," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[24] P. Shen, X. Lu, X. Hu, N. Kanda, M. Saiko, and C. Hori, "The NICT ASR System for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[25] X. Wang, A. Finch, M. Utiyama, T. Watanabe, and E. Sumita, "The NICT Translation System for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[26] K. Sudoh, G. Neubig, K. Duh, and K. Hayashi, "NTT-NAIST Syntax-based SMT Systems for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[27] K. Wolk and K. Marasek, "Polish - English Speech Statistical Machine Translation Systems for the IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[28] J. Wuebker, S. Peitz, A. Guta, and H. Ney, "The RWTH Aachen Machine Translation Systems for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[29] P. Bell, P. Swietojanski, J. Driesen, M. Sinclair, F. McInnes, and S. Renals, "The UEDIN ASR Systems for the IWSLT 2014 Evaluation," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[30] A. Birch, M. Huck, N. Durrani, N. Bogoychev, and P. Koehn, "Edinburgh SLT and MT System Description for the IWSLT 2014 Evaluation," in *Proceedings of*

the 11th International Workshop on Spoken Language Translation (IWSLT), Lake Tahoe, CA, 2014.

[31] R. W. M. Ng, M. Doulaty, R. Doddipatla, W. Aziz, K. Shah, O. Saz, M. Hasan, G. Alharbi, L. Specia, and T. Hain, "The USFD SLT system for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[32] S. Wang, Y. Wang, J. Li, Y. Cui, and L. Dai, "The USTC Machine Translation System for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, 2014.

[33] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, Phuket, Thailand, September 2005, pp. 79–86.

[34] H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning, "A conditional random field word segmenter," in *Fourth SIGHAN Workshop on Chinese Language Processing*, 2005.

[35] M. Federico, A. Cattelan, and M. Trombetti, "Measuring user productivity in machine translation enhanced computer assisted translation," in *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*, 2012. [Online]. Available: http://www.mt-archive.info/AMTA-2012-Federico.pdf

[36] S. Green, J. Heer, and C. D. Manning, "The efficacy of human post-editing for language translation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 439–448.

[37] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of the The Seventh Conference of the Association for Machine Translation in the Americas (AMTA)*, Cambridge, USA, 2006, pp. 223–231.

[38] E. W. Noreen, *Computer Intensive Methods for Testing Hypotheses: An Introduction*. Wiley Interscience, 1989.

[39] N. Chinchor, L. Hirschman, and D. D. Lewis, "Evaluating message understanding systems: An analysis of the third message understanding conference (muc-3)," *Computational Linguistics*, vol. 19, no. 3, pp. 409–449, 1993.

[40] S. Riezler and J. T. Maxwell, "On some pitfalls in automatic evaluation and significance testing for MT," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 57–64. [Online]. Available: http://www.aclweb.org/anthology/W/W05/W05-0908

[41] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

Appendix A. Automatic Evaluation

"case+punc" evaluation : case-sensitive, with punctuations tokenized
"no_case+no_punc" evaluation : case-insensitive, with punctuations removed

A.1. Official Testset (*tst2014*)

· All the sentence IDs in the IWSLT 2014 test set were used to calculate the automatic scores for each run submission.
· ASR and MT systems are ordered according to the *WER* and *BLEU* metrics, respectively.
· All automatic evaluation metric scores are given as percent figures (%).

TED : ASR English (ASR$_{EN}$)

System	WER	(# Errors)
NICT	**8.4**	**(1,831)**
EU-BRIDGE	9.8	(2,138)
MITLL-AFRL	9.9	(2,153)
KIT	11.4	(2,475)
FBK	11.4	(2,492)
LIUM	12.3	(2,689)
UEDIN	12.7	(2,763)
IOIT	19.7	(4,283)

TED : ASR German (ASR$_{DE}$)

System	WER	(# Errors)
KIT	**24.0**	**(5,660)**
UEDIN	35.7	(8,438)
FBK	38.8	(9,167)

TED : ASR Italian (ASR$_{IT}$)

System	WER	(# Errors)
VECSYS-LIUM	**21.9**	**(5,165)**
MITLL-AFRL	23.0	(5,440)
FBK	23.8	(5618)
KIT	25.4	(5,997)

TED : SLT English-French (SLT$_{EnFr}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
KIT	**27.45**	57.80	**28.16**	56.87
RWTH	26.94	57.29	27.74	**56.22**
LIUM	26.82	59.03	27.85	57.69
UEDIN	25.50	**57.23**	26.26	56.24
FBK	25.39	59.53	26.11	58.57
LIMSI	25.18	60.70	25.88	59.69
USFD	23.45	59.94	24.14	58.97

TED : SLT English-German (SLT$_{EnDe}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
KIT	**17.05**	**68.01**	**17.58**	**66.97**
UEDIN	17.00	68.36	17.51	67.30
USFD	14.75	70.15	15.24	69.15
KLE	13.00	71.70	13.64	70.33

TED : SLT German-English (SLT$_{DeEn}$)

System	*case sensitive*		case insensitive	
	BLEU	TER	BLEU	TER
EU-BRIDGE	**19.09**	**63.80**	**19.59**	**62.94**
KIT	18.34	63.91	18.85	62.99
UEDIN	17.67	66.04	18.18	65.12
RWTH	17.24	65.04	17.78	64.07
KLE	9.95	74.05	10.36	72.97

TED : MT English-French (MT$_{EnFr}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
EU-BRIDGE	**36.99**	45.20	**37.85**	**44.32**
KIT	36.22	**45.18**	36.97	44.37
UEDIN	35.91	45.78	36.64	45.04
RWTH	35.72	44.54	36.46	43.77
MITLL-AFRL	35.48	45.69	36.90	44.49
FBK	34.24	46.75	34.85	46.04
BASELINE	30.55	49.66	31.13	49.00
MIRACL	25.86	54.16	26.97	53.02
SFAX	16.09	62.89	17.33	61.48

TED : MT English-German (MT$_{EnDe}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
EU-BRIDGE	**23.25**	**57.27**	**24.06**	**56.15**
KIT	22.66	57.70	23.35	56.66
UEDIN	22.61	58.95	23.14	57.92
NTT-NAIST	22.09	57.60	22.63	56.65
KLE	19.26	61.36	19.75	60.48
BASELINE	18.44	61.89	18.92	61.02

TED : MT German-English (SLT$_{DeEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
EU-BRIDGE	**25.77**	**54.61**	**26.36**	**53.76**
RWTH	25.04	55.49	25.61	54.65
KIT	24.62	55.62	25.16	54.77
NTT-NAIST	23.77	56.43	24.52	55.49
UEDIN	23.32	57.50	24.06	56.55
FBK	20.52	63.37	21.77	60.66
KLE	19.31	63.88	20.60	61.38
BASELINE	17.50	65.56	18.61	63.08

TED : MT English-Arabic (MT$_{EnAr}$)

System	BLEU	TER
UEDIN	**13.24**	**69.16**
KIT	13.05	71.62
BASELINE	11.12	72.88

TED : MT Arabic-English (MT$_{ArEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
MITLL-AFRL	**27.52**	**54.54**	**28.41**	**53.44**
UEDIN	25.46	57.07	26.22	56.02
BASELINE	19.88	63.30	20.48	62.31

TED : MT English-Spanish (MT$_{EnEs}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**35.63**	**45.10**	**36.47**	**44.12**
BASELINE	31.26	48.43	31.95	47.48

TED : MT Spanish-English (MT$_{EsEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**37.29**	**43.73**	**38.07**	**42.85**
BASELINE	33.31	46.07	33.80	45.38

TED : MT English-Farsi (MT$_{EnFa}$)

System	BLEU	TER
BASELINE	**6.48**	**81.14**

TED : MT Farsi-English (MT$_{FaEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
MITLL-AFRL	**18.37**	**66.02**	**19.03**	**65.03**
UEDIN	16.94	72.66	17.52	71.66
BASELINE	16.22	72.13	16.72	71.05

TED : MT English-Hebrew (MT$_{EnHe}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
BASELINE	**15.69**	**65.62**	**15.69**	**65.62**

TED : MT Hebrew-English (MT$_{HeEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**26.58**	**56.99**	**27.14**	**56.25**
BASELINE	23.66	58.66	24.20	57.83

TED : MT English-Polish (MT$_{EnPl}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
PJIIT	**16.10**	**74.82**	**16.60**	**73.64**
BASELINE	9.75	82.60	10.16	81.44
LIA	7.79	86.89	10.12	82.31

TED : MT Polish-English (MT$_{PlEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
PJIIT	**18.33**	**65.60**	**18.96**	**64.59**
BASELINE	13.94	68.75	14.49	67.63

TED : MT English-Portuguese (MT$_{EnPt}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**32.41**	**45.85**	**33.12**	**44.87**
BASELINE	31.32	47.06	31.97	46.19

TED : MT Portuguese-English (MT$_{PtEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
BASELINE	**35.78**	**43.13**	**36.16**	**42.61**
UEDIN	34.66	46.11	35.28	45.52

TED : MT English-Russian(MT$_{EnRu}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	**11.21**	**73.15**	**11.21**	**72.24**

TED : MT English-Slovenian(MT$_{EnSl}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
LIA	**10.36**	**71.81**	**12.69**	**67.80**
BASELINE	8.53	73.75	8.87	72.76

TED : MT English-Turkish(MT$_{EnTr}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	**6.97**	**79.93**	**7.36**	**78.65**
UMONTREAL	4.76	80.67	5.51	79.28

TED : MT English-Chinese(MT$_{EnZh}$)

System	*character-based*	
	BLEU	TER
USTC	21.64	65.71
KIT	18.31	66.43
HKUST	16.41	74.35
BASELINE	15.56	80.48
UMONTREAL	7.40	81.89

TED : MT Russian-English (MT$_{RuEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
MITLL-AFRL	**19.30**	**63.95**	**20.22**	**62.64**
BASELINE	15.48	69.93	15.95	68.91

TED : MT Slovenian-English (MT$_{SlEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	**13.69**	**70.79**	**14.07**	**69.83**

TED : MT Turkish-English (MT$_{TrEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	**12.52**	**76.96**	**13.10**	**75.77**

TED : MT Chinese-English (MT$_{ZhEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
USTC	**15.65**	**69.65**	**16.35**	**68.62**
NICT	14.05	71.68	14.88	70.42
MITLL-AFRL	12.83	74.74	13.51	73.58
BASELINE	11.22	72.43	11.79	71.37
HKUST	9.64	76.67	10.83	74.16

A.2. Progress Test Set (*tst2013*)

· All the sentence IDs in the IWSLT 2013 test set were used to calculate the automatic scores for each run submission.
· ASR and MT systems are ordered according to the *WER* and *BLEU* metrics, respectively.
· For each task, the best score of each metric is marked with **boldface**.
· All automatic evaluation metric scores are given as percent figures (%).

TED: ASR English tst2013

System	IWSLT 2013 WER	IWSLT 2013 (# Errors)	IWSLT 2014 WER	IWSLT 2014 (# Errors)
NICT	**13.5**	**(5,734)**	**10.6**	**(4,518)**
MITLL-AFRL	15.9	(6,788)	13.7	(5,856)
KIT	14.4	(6,115)	14.2	(6,044)
FBK	23.2	(9,899)	14.7	(6,247)
LIUM	—		16.0	(6,818)
UEDIN	22.1	(9,413)	16.3	(6,963)
IOIT	27.2	(11,578)	24.0	(10,206)

TED: ASR German tst2013

System	IWSLT 2013 WER	IWSLT 2013 (# Errors)	IWSLT 2014 WER	IWSLT 2014 (# Errors)
KIT	**25.7**	**(4,932)**	**25.4**	**(5,885)**
UEDIN	37.8	(7,250)	35.0	(6,720)
FBK	37.5	(7,199)	37.8	(7,261)

TED : MT English-French test 2013(MT$_{EnFr}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
EU-BRIDGE	**40.50**	43.27	**41.65**	42.06
KIT	40.12	43.09	41.11	42.04
RWTH	39.72	**42.58**	40.73	**41.52**
UEDIN	39.59	43.80	40.45	42.78
MITLL-AFRL	39.08	44.05	40.59	42.73
FBK	38.20	44.83	38.99	43.88
BASELINE	33.20	48.91	33.81	48.07
MIRACL	29.63	51.96	30.91	50.65

TED : MT English-German test 2013 (MT$_{EnDe}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
EU-BRIDGE	**26.22**	**53.62**	**27.30**	**52.34**
KIT	26.03	53.83	26.77	52.81
NTT-NAIST	25.80	53.86	26.55	52.75
UEDIN	25.33	55.12	26.13	53.93
KLE	21.69	58.27	22.25	57.32
BASELINE	20.96	58.48	21.52	57.58

TED : MT German-English test 2013 (MT$_{DeEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
EU-BRIDGE	**28.77**	**50.52**	**29.29**	**49.63**
KIT	27.98	50.92	28.55	50.04
NTT-NAIST	27.81	51.62	28.32	50.82
UEDIN	27.60	52.43	28.26	51.44
RWTH	27.59	51.33	28.08	50.41
FBK	25.45	55.80	26.07	54.88
KLE	23.59	57.38	24.18	56.47
BASELINE	20.26	60.33	20.89	59.48

TED : MT English-Arabic test 2013(MT$_{EnAr}$)

System	BLEU	TER
UEDIN	**14.20**	**65.97**
KIT	14.15	68.29
BASELINE	12.68	68.94

TED : MT Arabic-English test 2013 (MT$_{ArEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
MITLL-AFRL	**31.48**	**49.88**	**32.41**	**48.76**
UEDIN	29.06	53.02	29.74	52.03
BASELINE	21.63	60.32	22.46	59.12

TED : MT English-Spanish test 2013 (MT$_{EnEs}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**34.74**	**45.75**	**35.42**	**44.78**
BASELINE	30.63	49.39	31.14	48.57

TED : MT Spanish-English test 2013(MT$_{EsEn}$)

System	*case sensitive* BLEU	*case sensitive* TER	*case insensitive* BLEU	*case insensitive* TER
UEDIN	**39.13**	**41.37**	**39.75**	**40.60**
BASELINE	34.18	44.63	34.70	44.00

TED : MT English-Farsi test 2013 (MT$_{EnFa}$)

System	BLEU	TER
BASELINE	7.05	78.90

TED : MT Farsi-English test 2013 (MT$_{FaEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
MITLL-AFRL	**19.47**	**63.27**	**20.11**	**62.27**
UEDIN	16.51	82.50	16.87	81.58
BASELINE	14.04	83.01	14.44	82.09

TED : MT English-Hebrew test 2013(MT$_{EnHe}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	**15.92**	**64.16**	**15.92**	**64.16**

TED : MT Hebrew-English test2013 (MT$_{HeEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
UEDIN	**29.70**	**52.40**	**30.51**	**51.35**
BASELINE	25.97	55.40	26.74	54.23

TED : MT English-Polish test2013 (MT$_{EnPl}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
PJIIT	**25.92**	**61.04**	**26.62**	**59.94**
BASELINE	11.12	75.95	11.67	74.78

TED : MT Polish-English test2013 (MT$_{PlEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
PJIIT	**27.99**	**58.01**	**28.61**	**57.10**
BASELINE	17.25	66.44	17.75	65.44

TED : MT English-Portuguese test 2013(MT$_{EnPt}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	31.38	46.42	31.89	45.66
UEDIN	33.20	44.90	33.93	43.90

TED : MT Portuguese-English test 2013 (MT$_{PtEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	37.85	40.87	38.26	40.35
UEDIN	37.34	42.91	37.80	42.30

TED : MT English-Russian test 2013(MT$_{EnRu}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	**14.01**	**70.47**	**14.01**	**69.44**

TED : MT Russian-English test 2012 (MT$_{RuEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
MITLL-AFRL	**24.30**	**57.59**	**25.39**	**56.25**
BASELINE	19.82	63.56	20.40	62.46

TED : MT English-Slovenian test 2013 (MT$_{EnSl}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	**9.63**	**73.32**	**9.97**	**72.34**

TED : MT Slovenian-English test2013 (MT$_{SlEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	**14.64**	**68.68**	**15.19**	**67.63**

TED : MT English-Turkish test 2013 (MT$_{EnTr}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	**6.85**	**80.40**	**7.21**	**79.08**
UMONTREAL	4.06	83.97	4.77	82.50

TED : MT Turkish-English test 2013 (MT$_{TrEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
BASELINE	**13.30**	**75.17**	**13.95**	**74.00**

TED : MT English-Chinese test2013 (MT$_{EnZh}$)

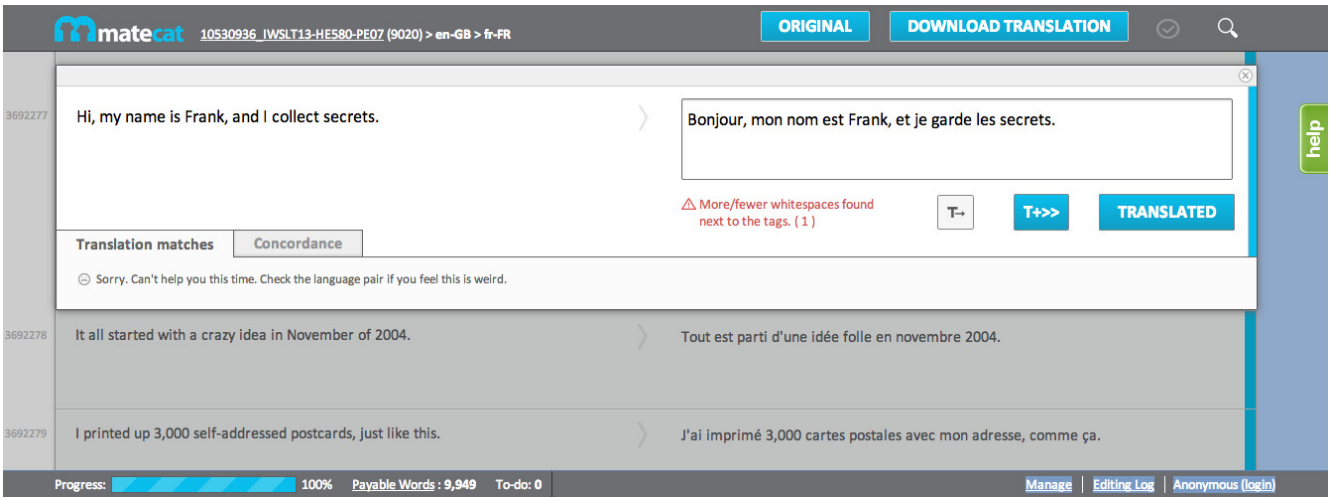
System	*character-based*	
	BLEU	TER
USTC	**22.49**	**63.74**
KIT	21.01	63.12
HKUST	18.81	70.94
BASELINE	18.23	76.15
UMONTREAL	7.93	80.47

TED : MT Chinese-English test 2013(MT$_{ZhEn}$)

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
USTC	**18.12**	**66.28**	**18.85**	**65.23**
NICT	16.57	67.96	17.36	66.76
MITLL-AFRL	15.59	70.89	16.32	69.68
BASELINE	13.40	68.85	14.00	67.90
HKUST	11.89	72.33	13.08	70.10

Appendix B. Human Evaluation

Interface used for the bilingual post-editing task



Post-editing instructions given to professional translators

In this task you are presented with automatic translations of TED Talks captions.

You are asked to post-edit the given automatic translation by applying the minimal edits required to transform the system output into a fluent sentence with the same meaning as the source sentence.

While post-editing, remember that the post-edited sentence is to be intended as a transcription of spoken language. Note also that the focus is the correctness of the single sentence within the given context, NOT the consistency of a group of sentences. Hence, surrounding segments should be used to understand the context but NOT to enforce consistency on the use of terms. In particular, different but correct translations of terms across segments should not be corrected.

Examples:

Source: This next one takes a little explanation before I share it with you.
Automatic translation: ...avant que je partage avec vous.
Post-editing 1: ...avant de le partager avec vous.
Post-editing 2: ...avant que je le partage avec vous. (preferred - minimal editing and acceptable in spoken language)

Source: And the table form is important.
Automatic translation: Et la forme de la table est importante.
Post-editing 1: La forme de la table est également importante.
Post-editing 2: Et la forme de la table est importante. (preferred - no editing - slightly less fluent but better fitting the source speech transcription)

Source: Everyone who knew me before 9/11 believes...
Automatic translation: ...avant le 11/9...
Post-editing 1: ...avant le 11 septembre...
Post-editing 2: ...avant le 11/9... (preferred - no editing - better fitting the source)

FBK @ IWSLT 2014 - ASR track

B. BabaAli[1], R. Serizel[2], S. Jalalvand[2], D. Falavigna[2], R. Gretter[2] and D. Giuliani[2]

[1]College of Science, University of Tehran, Tehran, Iran
[2]HLT research unit, Fondazione Bruno Kessler (FBK), Trento, Italy
babaali@ut.ac.ir (giuliani, falavi, gretter)@fbk.eu

Abstract

This paper reports on the participation of FBK in the IWSLT 2014 evaluation campaign for Automatic Speech Recognition (ASR), which focused on the transcription of TED talks. The outputs of primary and contrastive systems were submitted for three languages, namely English, German and Italian.

Most effort went into the development of the English transcription system. The primary system is based on the ROVER combination of the output of 5 transcription subsystems which are all based on the Deep Neural Network - Hidden Markov Model (DNN-HMM) hybrid. Before combination, word lattices generated by each sub-system are rescored using an efficient interpolation of 4-gram and Recurrent Neural Network (RNN) language models. The primary system achieves a Word Error Rate (WER) of 14.7% and 11.4% on the 2013 and 2014 official IWSLT English test sets, respectively. The subspace Gaussian mixture model (SGMM) system developed for German achieves 39.5% WER on the 2014 IWSLT German test sets. For Italian, the primary transcription system was based on hidden Markov models and achieves 23.8% WER on the 2014 IWSLT Italian test set.

1. Introduction

This paper describes the English, German, Italian FBK large vocabulary continuous speech recognition systems developed for the IWSLT 2014 evaluation campaign (*http://workshop2014.iwslt.org*). As the IWSLT 2013 evaluation campaign [1], the ASR track of the IWSLT 2014 evaluation campaign focused on the transcription of TED talks (*http://www.ted.com*). The main challenges for automatic transcriptions of TED talks include: variability in acoustic conditions, large variability of topics (hence a large, unconstrained vocabulary), presence of non-native speakers and a rather informal speaking style.

Most effort went into the development of the English transcription system. The primary system for English is based on the ROVER combination [2] of the output of 5 transcription sub-systems. Most of the progress demonstrated for English, w.r.t. the FBK participation into the IWSLT 2013 campaign [3], is due to the switching from the Hidden Markov Model - Gaussian Mixture Model (HMM-GMM) approach to DNN-HMM hybrid systems, the use of an improved n-gram language model, and an N-best list rescoring strategy based on an interpolation of n-gram and RNN Language Models (LMs). In addition, we took advantage by using the Kaldi open source toolkit for system development [4].

In this paper, more details are reported for the experiments conducted for English than for German and Italian.

The rest of this paper is organized as follow. Section 2 describes the speaker diarization module, while Section 3 describes the ASR systems developed for English and Section 4 describes the ASR systems developed for German and Italian. Section 5 presents the automatic transcription results achieved on the TED talk data for all languages. Finally, some conclusions are reported in Section 6.

2. Speaker diarization

The input audio signal is first processed by a speaker diarization module which performs: start-end point detection, speech segment classification and segment clustering based on Bayesian information criterion [5]. At the end of this process, each audio file has assigned a set of temporal segments, each having associated a label that indicates the cluster to which it belongs (e.g. female_1, male_1, etc). This processing is common to all transcription systems presented in this paper and was not changed since the IWSLT 2013 evaluation [3].

3. English Transcription System

3.1. Acoustic data selection

Acoustic Model (AM) training was performed using in-domain data. To this end, TED talk videos released before the cut-off date, 31 December 2010, were downloaded with the corresponding subtitles which are not a verbatim transcription of the speech. Subtitles are, in fact, content-only transcriptions in which anything irrelevant to the content is ignored, including most non-verbal sounds, false starts, repetitions, incomplete or revised sentences and superfluous speech by the speaker. A simple but robust automatic procedure was implemented to select only audio data with an accurate transcription. The approach adopted is that of se-

This work was done while Bagher BabaAli was at FBK as a Visiting Researcher.

lecting only those portions in which the human transcription and an automatic transcription agree [6]. For details on the speech data selection procedure adopted the reader can refer to [3].

The collected data consisted in 820 TED talks, for a total duration of ∼216 hours, with ∼166 hours of actual speech. The speech data selection procedure resulted in ∼144 hours of transcribed speech effectively used for AM training. This year, for acoustic model training we used only this in-domain data, while the previous year, in-domain data was augmented with HUB4 training data [3].

3.2. LM training

Text data used for training the LMs are those released for the IWSLT2013-SLT Evaluation Campaign. Before training, texts were cleaned, normalized (punctuation was removed, numbers and dates were expanded) and double lines were removed. Training documents come from the following three sources:

- **giga5** GIGAWORD 5-th edition. Contains documents stemming from seven distinct international sources of English newswire. It is released from the Linguistic Data Consortium (see *http://www.ldc.upenn.edu*). In total it contains about 4G words.

- **wmt13** Formed by documents in WMT12 news crawl, news commentary v7 and Europarl v7 (see IWSLT2013 official web site for some more details about these corpora). In total it contains about 1G words.

- **ted13** An in-domain set of texts extracted from TED talks transcriptions. It contains about 2.7M words.

Three 4-gram LMs, namely **giga5**, **wmt13** and **ted13** were independently trained on the three sources using the modified shift-beta smoothing method as supplied by the IRSTLM toolkit [7]. Then, two additional "mixture" LMs were trained using the "mix" adaptation method implemented in the IRSTLM toolkit [7]. The **two-mix** LM is built mixing the smoothed (with the modified shift-beta approach) n-grams of both **wmt13** and **ted13** collections, the **all-mix** LM is obtained mixing the smoothed (with improved Kneser-Ney method [8]) n-grams of all of the three collections aforementioned: **giga5**, **wmt13** and **ted13**. We point out that in the case of **all-mix** LM training no pruning of singleton 4-grams was applied.

A further 4-gram LM, namely **sel172M**, was trained on 172M words automatically selected from **giga5** collection in order to match the in-domain set of documents **ted13**. Also in this case the IRSTLM toolkit was employed, together with the modified shift beta method for smoothing probabilities of the n-grams not seen in the training set. The method used for automatically selecting documents from the **giga5** collection is based on "term frequency inverse documents frequency" (TFIDF) coefficients and uses the **ted13** collection as seed corpus. Details can be found in [9].

Finally, two different RNN LMs (namely **RNNLM1** and **RNNLM2**) were trained, using the toolkit described in [10], on the **ted13** collection and on a text corpus including both the **ted13** collection and a subset of documents (containing around 10M words) automatically extracted from the **giga5** collection, respectively. Hence, the **RNNLM2** LM was trained over around 12.7 M words, mapping the singletons into the "<unk>" symbol. The RNNLM1 LM has 450 hidden neurons in its hidden layer and the RNNLM2 LM has 500 hidden neurons.

Note that, the **wmt13** LM is the LM used by ASR systems developed for the IWSLT 2013 evaluation, while the **two-mix** LM is used by all ASR systems developed for the IWSLT 2014 evaluation. Perplexity (PP) and out-of-vocabulary (OOV) rates measured on the reference transcriptions of the IWSLT English 2010 development data set (containing 44505 words) are reported in Table 1. We can see that the **two-mix** LM exhibits a significant lower perplexity that the **wmt13** LM. Column "Interp." in the table reports PP and OOV rate obtained by linearly interpolating the **all-mix**, **ted13**, **sel172M**, **RNNLM1** and **RNNLM2** LMs: interpolation weights are estimated in order to minimize the overall perplexity on the transcriptions of the English 2010 development set. Interpolation of these LMs is applied at recognition time for N-best list rescoring, as it will be detailed in Section 3.4.2.

LM	giga5	wmt13	ted13	two-mix	Interp.
PP	495	461	223	378	289
%OOV	0.4	1.7	7.5	1.6	0.3

Table 1: Perplexities and % OOV rates measured with several LMs on transcriptions of IWSLT English 2010 development data set.

3.3. Lexicon

Word pronunciations in the English lexicon are based on a set of 45 phones. They were generated by merging different source lexica for American English (LIMSI '93, CMU dictionary, Pronlex). In addition, phonetic transcriptions for a number of missing words were generated by using the phonetic transcription module of the Festival speech synthesis system. The lexicon did not change with respect to the previous year.

3.4. ASR system development using Kaldi

In the open source software Kaldi [4], there are two separate setups for neural network training implementation, namely Dan's and Karel's setups or recipes [11, 12]. In both of these setups, the last (output) layer is a softmax layer whose output dimension equals the number of context-dependent states of a pre-trained HMM-GMM system. The neural net is trained

to predict the posterior probability of each context-dependent HMM state [13, 14]. During decoding the posterior probabilities are divided by the prior probability of each state to form a pseudo-likelihood that is used in place of the state emission probabilities in the HMM. Depending on which of the two setups is used the performance is different because of many differences in the recipes. For example, Karel's setup uses pre-training but Dan's setup does random initialization; Karel's setup uses early stopping using a validation set but Dan's setup uses a fixed number of epochs and averages the parameters over the last few epochs of training. Many other aspects of the training procedure are also different (nonlinearity types, learning rate schedules, etc.). Two speaker-adaptive DNN-HMM systems were developed by using the Dan's and Karel's setups.

3.4.1. Acoustic modeling

For acoustic modeling 13 mel-frequency cepstral coefficients (MFCCs), including the zero order coefficient, are extracted from the signal every 10ms by using a Hamming window of 25ms length. These features are then mean/variance normalized on a speaker-by-speaker basis, spliced by +/- 3 frames next to the central frame and projected down to 40 dimensions using linear discriminant analysis (LDA) and Maximum Likelihood Linear Transform (MLLT). A single feature-space Maximum Likelihood Linear Regression (fMLLR) transform for each training speaker is then estimated and applied to train speaker-adaptively trained (SAT) triphone HMMs. These SAT triphone HMM have 6,349 tied-states and 130,000 Gaussians. The speaker-adaptive DNN-HMM hybrid systems are built on top of LDA-MLLT-fMLLR features and SAT triphone HMMs.

A first DNN is trained using the Karel's setup. An eleven frames context window of LDA-MLLT-fMLLR features (5 frames at each side) is used as input to form 440 dimensional feature vector. The DNN have 6 hidden layers each with 2048 neurons, the resulting architecture can be summarized as follows: 440x2048x2048x2048x2048x2048x2048x6349. The DNN is trained in several stages including Restricted Boltzmann Machines (RBM) pre-training, mini-batch Stochastic Gradient Descent training, and sequence-discriminative training such as Minimum Phone Error (MPE) and state-level Minimum Bayes Risk (sMBR).

A second DNN is trained based on the Dan's setup. A nine frames context window of LDA-MLLT-fMLLR features (4 frames at each side) is used as input to form 360 dimensional feature vectors. The DNN is a p-norm DNN with 5 hidden layers and p-norm (input, output) dimensions of (4000, 400) respectively, i.e. the nonlinearity reduces the dimension tenfold [12]. 12000 sub-classes are used, and the number of parameters is 11.0 million. The Dan's setup does not support RBM pretraining. Instead it performs something similar to the greedy layer-wise supervised training [15] or the layer-wise backpropagation of [14]. The network is initialized randomly with one hidden layer, trained for a short time (typically less than an epoch, meaning less than one full-pass through the data), then the layer of weights that go to the softmax layer is removed, a new hidden layer and two sets of randomly initialized weights are added, and trained again. This is repeated until we have four layers. The initial and final learning rates in our training setup are 0.08 and 0.0008 respectively, and during training is decreased exponentially, except for a five epochs at the end during which it is kept fixed. Dan's setup was originally written to support parallel training on multiple CPUs or GPUs. During training, a data-parallel method based on a periodic averaging the parameters of separate Stochastic Gradient Descent runs.

3.4.2. Decoding process

At recognition stage, LDA-MLLT-fMLLR features are first generated by using auxiliary HMMs. To this end, a decoding pass with speaker-independent GMM-HMM is conducted to produce a word lattice for each utterance. A single fMLLR transform for each speaker is then estimated from sufficient statistics collected from word lattices with respect to SAT triphone HMMs. These transforms are hence used in the second decoding pass with SAT HMM to produce new word lattices. A second set of fMLLR transforms is estimated from new word lattices and combined with the first set of transforms. Then a decoding pass is conducted on the obtained fMLLR adapted acoustic features with the DNN-HMM hybrid system, where the DNN is trained to provide posterior probability estimates for the SAT triphone HMM tied-states.

All decoding passes make use of a decoding graph built using a "pruned" version of the **two-mix** LM introduced above. The word lattice generated for each utterance by the DNN-HMM hybrid system is rescored with the "non pruned" **two-mix** LM in order to produce the final ASR hypothesis. Alternatively, as mentioned in Section 3.2, N-best (N=100) list is generated and rescored. In this case, rescoring consists in recomputing, for each hypothesis in the list, the corresponding LM probability as a linear interpolation of the probabilities given by the **all-mix**, **ted13**, **sel172M**, **RNNLM1** and **RNNLM2** LMs. Its worthwhile to mention that the interpolation weights are estimated in order to minimize the perplexity over all the 1-best hypotheses.

3.5. Development of complementary ASR systems

In view of system combination with ROVER, we explored the way to develop complementary systems. To this end, acoustic models of the HMM-GMM system for the IWSLT 2013 ASR English evaluation [3] were used to provide tied-state alignment to train two additional DNN-HMM hybrid systems which are described below.

3.5.1. Two-pass HMM-GMM system

The two decoding pass HMM-GMM system developed for the IWSLT 2013 evaluation uses the **wmt13** LM [3]. A first complementary systems developed for the IWSLT 2014 eval-

uation is obtained using the **two-mix** LM instead.

3.5.2. DNN-HMM systems

The first DNN-HMM system was trained on the tied-state alignment obtained with the SAT triphone HMMs used in the first decoding pass by the 2013 HMM-GMM system. The DNN was however trained on unnormalized acoustic features. The second DNN-GMM system was trained on the tied-state alignment obtained with the SAT triphone HMMs used in the second decoding pass by the 2013 HMM-GMM system and on SAT features. At recognition stage, a two pass DNN-HMM decoding system is obtained when word transcriptions generated by the DNN-HMM system using unnormalized acoustic features are used to supervise the extraction of the SAT acoustic features for a second decoding pass with the SAT DNN-HMM system.

First-pass DNN-HMM
The first DNN is trained on 13 MFCC, including the zero coefficient, without speaker normalization. A 31-frame context window is applied, the 403-dimensional features vector is then decorrelated with discrete cosine transform (DCT) and projected on a 208-dimensional feature vector. Average and covariance normalisations are applied to this later feature vector and the resulting, normalized, vector is used as input to the DNN. The DNN is composed of 5 hidden layers with 1500 elements per layer. The DNN is trained with cross-entropy on 10021 triphone tied-states targets obtained from time alignment with the first pass models of the 2013 HMM-GMM system. The resulting architecture can be summarized as follows: 208x1500x1500x1500x1500x1500x10021.

The TNet software package [16] is used for training. The training set for the DNN is composed only of TED data as explained above. The training set is split into two sets with non-overlapping speaker: training (90%) and cross-validation (10%). The DNN weights are initialized randomly and pre-trained with RBM [17, 18]. The first layer is pre-trained with a Gaussian-Bernoulli RBM trained during 10 iterations with a learning rate of 0.005. The following layers are pre-trained with a Bernoulli-Bernoulli RBM trained during 5 iterations with a learning rate of 0.05. Mini-batch size is 500. For the back propagation training the learning rate is kept to 0.08 as long as the frame accuracy on the cross-validation set progresses by, at least, 0.5% between successive epochs. The learning rate is then halved at each epoch until the frame accuracy on the cross-validation set fails to improve by at least 0.1%. The mini-batch size is 1024. In both pre-training and training, a first-order momentum of 0.9 is applied.

Second-pass SAT DNN-HMM
The second DNN is trained on the 39 SAT features as generated for the second pass triphone HMM of the 2013 HMM-GMM system. A 31-frame context window is applied. The resulting 1209-dimensional features vector is decorrelated with DCT and projected on a 468-dimensional feature vector. Average and covariance normalization is applied and the resulting, normalized, vector is used as input to the DNN. The DNN is composed of 5 hidden layers with 1500 elements per layer. It is trained with cross-entropy on 10021 triphone tied-states targets obtained from time alignment with the second pass models of the HMM-GMM baseline. The resulting architecture can be summarized as follows: 468x1500x1500x1500x1500x1500x10021. The training was conducted following the same set up as for the first-pass DNN above.

4. German and Italian transcription systems

For this evaluation, we decided to focus our efforts mostly on English and to dedicate a limited attention to German and Italian. For both languages we wanted to compare our in-house proprietary system with the Kaldi recognizer, but due to the aforementioned limitations, at the end we did the following submissions:

- **Italian primary** in-house SAT HMM-GMM system (see [3] for details);
- **Italian contrastive1** SAT Subspace Gaussian Mixture Model (SGMM) system developed with Kaldi [4];
- **German primary** SAT SGMM system developed with Kaldi.

4.1. Acoustic data

Concerning *Italian*, we could use the following corpora:

- **Euronews Italian Data** provided by the organizers, amounting to about 76h:38m of reliable speech. The corresponding transcription was obtained after a further step of light supervision training, using the domain dependent AMs trained on the originally provided data.
- **Italian Internal data**: about 216h:31m of reliably transcribed (partly manually, partly with light supervision techniques) speech collected in the previous years and belonging to 3 domains: *Apasci*, a phonetically balanced corpus; *Italian Parliament* recordings, *TV news* recorded from RAI. All this data were recorded before June 30th, 2011.

This data amounted to slightly more than 293 hours, but in order to speed up Kaldi experiments we decided to sample the data, by keeping only the first 100 sentences for each audio file. This resulted in about 154h:19m of speech (74h:30m Euronews data + 79h:49m Internal data).

Instead, the in-house proprietary system was trained on the Italian Internal data only (216h:31m).

Concerning *German*, we could use the following corpora:

- **Euronews German data** provided by the organizers, amounting to about 72h:18m of reliable speech. The

corresponding transcription was obtained after a further step of light supervision training, using the domain dependent AMs trained on the originally provided data.

- **German WEB data**: about 158h:47m of speech data transcribed using light supervision techniques, collected before July 2012.

The effective material used for training consisted in about 206h:54m of speech (66h:45m Euronews data + 140h:09m German WEB data).

4.2. Textual data

To build the **German LM** we used text data from various sources, including Europarl data, news from 2005 to June 30th, 2012, and of course the **ASR LM Training Data German** provided by IWSLT organizers. The total amount of words was about 1,130 million words. These data were processed in order to perform a normalization including in particular number and compound words splitting, which was performed in a fully automatic way described in [3]. After normalization, a 4-grams language model was built, resulting in about 481.3 millions of 4-grams. A pruned version of this LM, including about 9,7 millions of 4-grams, was used to build the FST used to build the lattices during decoding, while the full LM was used to rescore the lattices. The lexicon was fixed to the most frequent 200K words; the phonetic transcription was generated by our in-house system.

To build the **Italian LM** we used text data coming from news collected from 2005 to June 30th, 2011, in addition to the **ASR LM Training Data Italian** provided by IWSLT organizers. The total amount of words was about 985 million words. After text normalization and number splitting, a 4-grams language model was built, resulting in about 427,6 millions of 4-grams. For the contrastive system using Kaldi, a pruned version of this LM, including about 6,5 millions of 4-grams, was used to build the FST used to build the lattices during decoding, while the full LM was used to rescore the lattices. For the primary in-house system a static FSN was built using a pruned version of the LM, including was built 15,3 million 4grams. In both cases, the lexicon was fixed to the most frequent 200K words; the phonetic transcription was generated by our in-house system.

4.3. Decoding process

Both for German and Italian, we performed a two stage recognition. For the two Kaldi SGMM systems (German primary and Italian contrastive1) the two stage recognition was followed by a linguistic rescoring stage, obtained using the full LM over the generated lattices.

For the in-house system (Italian primary), no final LM rescoring was performed. Details about the AM adaptation performed for the second step decoding are described in [3].

5. Recognition Experiments

5.1. Results on English TED talks

Recognition experiments were carried out on the IWSLT 2014 English ASR development and evaluation data sets listed in Table 2. These data sets were released over several IWSLT evaluation campaigns. Recognition experiments on dev2012, tst2013 and tst2014 were always conducted in fully automatic mode. Instead, recognition experiments on all the other data sets (dev2010, tst2011 and tst2012) were conducted exploiting the provided manual segmentation.

Data Set	N. of Talks	Duration
dev2010	19	4h:00m
tst2011	8	1h:07m
dev2012	10	1h:57m
tst2012	11	1h:45m
tst2013	28	4h:38m
tst2014	15	2h:24m

Table 2: Details of the IWSLT 2014 English ASR development (dev) and evaluation (tst) data sets.

As a reference, Table 3 reports results achieved with the 2013 HMM-GMM system [3]. Column "Pruned LM" gives results obtained by the second decoding pass (see Section 3.5.1) using a pruned version of the LM, that is the **wmt13** LM introduced in Section 3.2. Column "Rover" in Table 3 reports results achieved with a combination, using ROVER, of 4 recognition outputs resulting from rescoring the word lattices generated by the second decoding pass by using different unpruned LMs [3]. The 23.7% WER reported on tst2013 data set is the result achieved by the FBK 2013 primary system in the IWSLT 2013 ASR evaluation campaign.

Data Set	System 2013	
	Pruned LM	Rover
dev2010	17.5	16.1
tst2011	15.6	13.6
dev2012	19.3	-
tst2012	17.6	16.1
tst2013	25.2	23.7

Table 3: % WER achieved by the HMM-GMM 2013 system on several English data sets. Results were obtained by: decoding with the pruned **wmt13** LM and performing ROVER combination of 4 different rescored outputs.

5.1.1. Experiments with Kaldi systems

Table 4 reports results with two SAT DNN-HMM systems developed with the Kaldi toolkit. "Dan" and "Karel" indicate the recipe, provided within the Kaldi toolkit, used to train the DNN.

Data Set	Kaldi DNN implementation	
	"Dan" (Pruned LM/Rescoring)	"Karel" (Pruned LM/Rescoring)
dev2010	14.8/13.4	13.4/12.5
tst2011	12.8/11.5	11.5/10.7
dev2012	18.0/17.0	16.3/15.3
tst2012	12.7/11.7	11.7/10.8
tst2013	19.4/18.0	17.5/16.4

Table 4: Results, in % WER, achieved by two different DNN-HMM systems on several English data sets. For each system and data set, it is reported the result achieved by: decoding with the pruned **two-mix LM** and performing rescoring of word lattices with the corresponding unpruned LM.

From results reported in Table 4 we can conclude that the "Karel" recipe allows to train a DNN which is consistently more effective than the DNN trained with the "Dan" recipe. In addition, performing rescoring of word lattices with the unpruned LM provides tangible benefit, for example dropping the WER, on the dev2010 data set, from 13.4% to 12.5% when using the the "Karel" DNN-HMM system.

The comparison of results reported in Tables 3 and 4, allows to appreciate the net improvements of the 2014 DNN-HMM systems over the 2013 HMM-GMM system. We believe that this major improvement can be attributed to the adoption of the deep learning paradigm for acoustic modeling, a better LM and a more comprehensive training procedure offered by the Kaldi development toolkit.

Data Set	Kaldi DNN implementation	
	"Dan" (N-best rescoring)	"Karel" (N-best rescoring)
dev2010	12.9	11.9
tst2011	10.7	10.0
dev2012	15.5	14.2
tst2012	11.0	10.4
tst2013	16.5	15.2

Table 5: % WER achieved by two different DNN-HMM systems on several English data sets by performing N-best (N=100) list rescoring using an interpolation of 4-gram and RNN LMs.

Table 5 reports results performing N-best list rescoring using an interpolation of 4-gram and RNN LMs, as described in section 3.4.2. By comparing these results with those in Table 4, we can notice the effectiveness of the N-best list rescoring method.

5.1.2. Experiments with complementary systems

Table 6 reports recognition results obtained with the 2014 HMM-GMM and DNN-HMM systems described in Section 3.5 without performing word lattice rescoring. Rows p1-GMM and p1-GMM+p2-GMM report results achieved performing one and two passes of decoding with the 2013 HMM-GMM system (see Section 3.5.1). Performing a single decoding pass 17.8% and 25.7% WER are achieved on the dev2010 and tst2013 data sets, respectively. While performing two decoding passes 16.3% and 23.4% WER are achieved on the dev2010 and tst2013 data sets, respectively. These latter results can be directly compared with the 17.5% and 25.2% WER, achieved by the 2013 HMM-GMM system as reported in the "Pruned LM" column of Table 3. The performance improvement can be attribute at the use of a better LM (that is **two-mix** Vs. **wmt13** LM). Results achieved performing one and two decoding passes with the DNN-HMM systems are reported in rows p1-DNN and p1-DNN+p2-DNN, respectively. We can see that performing a single decoding pass 16.5% and 21.9% WER are achieved on the dev2010 and tst2013 data sets, respectively. While performing two decoding passes 15.4% and 20.7% WER are achieved on the dev2010 and tst2013 data sets, respectively. These results confirm, once again, the effectiveness of the DNN-HMM hybrid approach. However, they are not as good as those obtained with DNN-HMM systems developed with the Kaldi toolkit and reported in Table 4 ("Pruned LM" condition).

Complementary System	dev2010	tst2013
p1-GMM	17.8	25.7
p1-GMM+p2-GMM	16.3	23.4
p1-GMM+p2-DNN	15.6	20.1
p1-DNN	16.5	21.9
p1-DNN+p2-GMM	15.5	22.0
p1-DNN+p2-DNN	15.4	20.7

Table 6: Results, in % WER, with different complementary system configurations on the dev2010 and tst2013 English data sets.

Table 6 reports also results obtained alternating recognition passes conducted with HMM-GMM and DNN-HMM systems. For example, row p1-DNN+p2-GMM reports results obtained performing the first pass with the DNN-HMM system and the second pass with the HMM-GMM system, this results in 15.5% and 22.0% WER on the dev2010 and tst2013 sets, respectively. In the following, we will refer to this system as "AltSystem1". One additional combination we have tried was as follows. The AltSystem1 was used to generate a word lattice which was acoustically rescored using the p2-DNN systems: we will refer to this system as "AltSystem2". The AltSystem2 resulted in 15.2% and 20.3% WER on the dev2010 and tst2013 data sets, respectively. The output of systems AltSystem1 and AltSystem2 were considered for system combination in the hope that they were different one each other enough.

Sub-systems	dev2010	tst2013
DNN-HMM "Dan"	12.9	16.5
DNN-HMM "Karel"1	11.9	15.3
DNN-HMM "Karel"2	11.9	15.2
AltSystem1	15.5	22.0
AltSystem2	15.2	20.3
	ROVER	
	11.7	14.7

Table 7: Results, in % WER, achieved by individual sub-systems, and performing ROVER-based system combination, on the dev2010 and tst2013 English data sets.

5.1.3. System combination

The 2014 primary system for English is based on the principle of system combination by means of ROVER. Table 7 reports recognition results achieved by the 2014 primary system, which combines the outputs of 5 sub-systems previously introduced. Results achieved by individual sub-systems are also reported. DNN-HMM "Karel"1 and DNN-HMM "Karel"2 denotes two sub-systems that differ only for the number of iterations in training of the corresponding DNN.

For the tst2013 data set we can see that an improvement of 0.5% WER is achieved with the ROVER combination w.r.t. the best sub-system entering in the combination: from 15.2% to 14.7% WER. The obtained 14.7% WER can be directly compared with the 23.7% WER obtained by the 2013 primary system on the same data (see Table 3). This represents a substantial improvement in terms of performance.

On the 2014 IWSLT English test set the official evaluation result achieved by the primary system is 11.4% WER, with an improvement of 0.7% WER w.r.t. the performance of the best sub-system entering in the ROVER combination, that is 12.1% WER.

5.2. Results on German TED talks

The subspace Gaussian mixture model system developed for German achieves 39.5% WER on the 2014 IWSLT German test sets.

5.3. Results on Italian TED talks

For Italian, the primary transcription system was based on hidden Markov models and achieves 23.8% WER on the 2014 IWSLT Italian test set. The contrastive1 transcription system, based on SGMM, achieves 24.6% WER.

6. Conclusions

In this paper we have presented the systems we developed for the participation in the IWSLT 2014 ASR evaluation campaign: we developed systems for the English, German and Italian ASR tracks.

For English, substantial progress, with respect to our primary system submission to IWSLT 2013 campaign [3], was demonstrated. This progress is due to the switching from the pure HMM-GMM approach to the adoption of DNN-HMM hybrid systems, the adoption of a better n-gram language model, and an N-best list rescoring strategy based on an interpolation of n-gram and RNN language models. In addition, we took advantage by using the Kaldi open source toolkit for system development.

7. Acknowledgements

This work was partially funded by the European project EU-BRIDGE, under the contract FP7-287658.

8. References

[1] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 10th IWSLT Evaluation Campaign," in *Proc. of the International Workshop on Spoken Language Translation (IWSLT)*, 2013.

[2] J. Fiscus, "A post-processing system to yield reduced error rates: recognizer output voting error reduction (ROVER)," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 1997, pp. 347–354.

[3] D. Falavigna, R. Gretter, F. Brugnara, D. Giuliani, and R. H. Serizel, "FBK@ IWSLT 2013-ASR track," in *Proc. of the International Workshop on Spoken Language Translation (IWSLT)*, 2013.

[4] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Veselỳ, "The Kaldi Speech Recognition Toolkit," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2011.

[5] M. Cettolo, "Segmentation, Classification and Clustering of an Italian Broadcast News Corpus," in *Proc. of Content-Based Multimedia Inf. Access Conf. (RIAO)*, Paris, France, 2000, pp. 372–381.

[6] L. Lamel, J.-L. Gauvain, and G. Adda, "Investigating Lightly Supervised Acoustic Model Training," in *Acoustics, Speech and Signal Processing, 2001, IEEE International Conference on*, Salt Lake City, UT, 2001.

[7] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models," in *Proc. of ICSLP*, Brisbane, Australia, September 2008, pp. 1618–1621.

[8] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech and Language*, vol. 4, no. 13, pp. 359–393, 1999.

[9] D. Falavigna and G. Gretter, "Focusing language models for automatic speech recognition," in *Proc. of the International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, HK, December 2012.

[10] T. Mikolov, S. Kombrink, L. Burget, J. H. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5528–5531.

[11] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence discriminative training of deep neural networks," in *Proc. of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH 2013)*, 2013, pp. 2345–2349.

[12] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of Deep Neural Networks with Natural Gradient and Parameter Averaging," in *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, San Diego, CA, USA, 2015, To appear.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[14] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in *Proc. of 12th Annual Conference of the International Speech Communication Association (INTERSPEECH 2011)*, 2011, pp. 437–440.

[15] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS)*, vol. 19, 2007, pp. 153–160.

[16] K. Veselỳ, L. Burget, and F. Grézl, "Parallel training of neural networks for speech recognition," in *Text, Speech and Dialogue*. Springer, 2010, pp. 439–446.

[17] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[18] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *The Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

The UEDIN ASR Systems for the IWSLT 2014 Evaluation

Peter Bell, Pawel Swietojanski, Joris Driesen,
Mark Sinclair, Fergus McInnes, Steve Renals

School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK
{peter.bell,p.swietojanski, fergus.mcinnes,s.renals}@ed.ac.uk,
jdriesen@staffmail.ed.ac.uk, m.sinclair-7@sms.ed.ac.uk

Abstract

This paper describes the University of Edinburgh (UEDIN) ASR systems for the 2014 IWSLT Evaluation. Notable features of the English system include deep neural network acoustic models in both tandem and hybrid configuration with the use of multi-level adaptive networks, LHUC adaptation and Maxout units. The German system includes lightly supervised training and a new method for dictionary generation. Our voice activity detection system now uses a semi-Markov model to incorporate a prior on utterance lengths. There are improvements of up to 30% relative WER on the `tst2013` English test set.

1. Introduction

This paper describes our system for automatic speech recognition (ASR) of TED talks, used in the 2014 evaluation campaign of the International Workshop on Spoken Language Translation. We describe both our English and German systems, although the development of the two was carried out separately.

This is the third year we have participated in the English ASR task. Our 2012 system [1] used tandem-GMM acoustic models, using deep neural networks (DNNs) to derive bottleneck features, incorporating out-of-domain data from multiparty meetings using the multi-level adaptive networks (MLAN) scheme [2]. In 2013 [3], we combined DNN systems in both tandem and hybrid configurations, again using the MLAN scheme. We also made extensive improvements to our language models, devoting substantial efforts to text normalisation, and data selection using the cross-entropy difference score proposed by [4]. These improvements led to a WER reduction from 12.4% to 10.2% on the `tst2011` progress test set.

This year, our final system features a system combination of several complementary systems built using the HTK and Kaldi toolkits. On the language modelling side, other than using a larger 4-gram model for final rescoring, there are very few changes from last year. This year's system does not employ recurrent neural network language models, as we were unable to obtain gains with the size of models used. On the acoustic modelling side, there are a number of new features: improved speaker adaptation for the DNNs with our recently proposed Learning Hidden Unit Contributions (LHUC) scheme [5]; the use of Maxout [6] and rectified linear units for the DNNs [7]; sequence training of some neural networks [8]; and the use of mixed-band training data. These features of the system are described in more detail in Section 3.

The German system is described separately in Section 4. For German, our major challenge is the lack of reliably-transcribed in-domain acoustic training data, and a good quality dictionary, neither of which we have access to. Like last year, we rely in bootstrapping a system from the German portion of the GlobalPhone corpus, using a biased language model method. We also use a new technique for dictionary expansion [9].

In the 2013 evaluation, ASR systems were required for the first time to operate without manually-supplied segmentation of the test data into utterances. We therefore used an automatic voice activity detection (VAD) based segmenter on the `tst2013` set as input to the ASR. We have since identified a number of problems with the baseline VAD system used in 2013, including a mismatch to the acoustic conditions, and a tendency to segment too tightly, leading to word deletions at sentence boundaries: we describe our work to address these problems in Section 2.

2. Voice activity detection

There were substantial changes to this year's VAD system, used for both English and German systems. After comparing the ASR performance with VAD-based segmentation on manually-segmented development data, we observed a reduction in performance compared to when the manual segmentations were used directly, even when local speech/silence decisions were generally correct. We hypothesised that this is because the utterances are often semantically segmented by human annotators, making them better-suited to language models trained on complete sentences. Additionally, in our system we observed an unfortunate trade-off between an over-sensitive segmenter which

This work was supported by the European Union under the FP7 projects inEvent (grant agreement 287872) and EU-Bridge (grant agreement 287658), and by EPSRC Programme Grant grant EP/I031022/1, *Natural Speech Technology*.

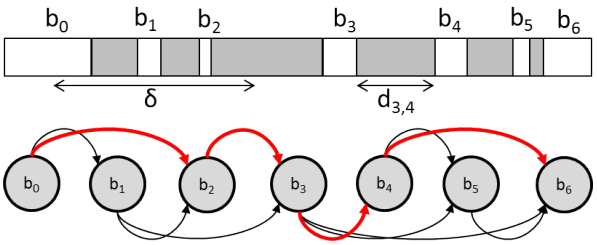


Figure 1: *An example of a candidate break sequence and associated state topology. The transitions highlighted in red show an example optimal break sequence* $B^* = \{b_0, b_2, b_3, b_4, b_6\}$

results in lots of short utterances, and an under-sensitive one, which can lead to excessively long segments or insertions and deletions at utterances boundaries.

As a solution to both problems, in [10], we proposed a novel technique based using a semi-Markov model with an prior on the duration of an utterance designed to yield segments more closely matching the distribution found in training data. For the prior, we used a log-normal distribution with parameters estimated on manually segmented training data. We found that the log-normal distribution generally provides a good fit to the distribution of utterance durations in the training data.

As input to the the semi-Markov decoder, we use a highly sensitive segmentation with small minimum duration constraint of 100ms. This produces many break points that would normally be detrimental to ASR if used directly. We decode this sequence of breaks using a semi-Markov decoder, to find the globally optimal sequence of breaks. The method is illustrated in Figure 1. Further details may be found in [10].

The initial segmentation is produced with an GMM-HMM based model. Speech and non-speech are modelled with diagonal covariance GMMs with 12 and 5 mixture components respectively. Features are calculated every 10ms from a 30ms analysis window and have a dimensionality of 14 (13 PLPs and energy). Models were trained on 70 hours of scenario meetings data from the AMI corpus using the provided manual segmentations as a reference.

3. English systems

3.1. Language modelling

Our language modelling setup is largely unchanged from last year, but we summarise it here for completeness. We trained standard Kneser-Ney smoothed n-gram language models on a combination of TED talk transcriptions as in-domain data, and out-of-domain data sources specified by the IWSLT rules. Table 1 shows the text data available, to which we applied substantial pre-processing and normalisation.

Following [4], we used all the available in-domain data,

Corpus	Total	Selected
TED	2.4M	2.4M
Europarl	53.1M	6.3M
News Commentary	4.4M	0.7M
News Crawl	693.5M	72.9M
Gigaword	2915.6M	232.9M
OOD total	3666.6M	312.8M

Table 1: Numbers of words in LM training sets.

Language model	Perplexity
TED 3-gram	183.2
OOD (312MW / 751MW) 3-gram	133.5 / 138.3
TED+OOD (312MW / 751MW) 3-gram	125.1 / 124.9
TED 4-gram	179.9
OOD (312MW / 751MW) 4-gram	123.9 / 126.4
TED+OOD (312MW / 751MW) 4-gram	114.9 / 113.4

Table 2: Perplexities of N-gram language models on TED development set.

and selected a subset of out-of-domain (OOD) data, D_s to minimise the cross-entropy difference:

$$D_S = \{s | H_I(s) - H_O(s) < \tau\} \quad (1)$$

where $H_I(s)$ is a cross-entropy of a sentence with a LM trained on in-domain data, $H_O(s)$ is a cross-entropy of a sentence with a LM trained on a random subset of the OOD data of similar size to the TED corpus, and τ is a threshold to control the size of D_S. Interpolation parameters were tuned on the `dev2010` and `tst2010` sets.

Table 2 shows perplexities of the in-domain, OOD and final interpolated LMs. In both Kaldi and HTK decoding pipelines the smaller 3-gram model was used for the primary decoding passes; when Kaldi's WFST-based decoder was used, the 3-gram was pruned to reduce memory requirements. In both cases, lattices were finally rescored using an unpruned 4-gram LM. Compared to 2013, when only models trained on 312MW set were used, this year we used the substantially larger 4-gram model trained on 715M words for the final pass. Due to the limitations of HDecode, we again limited the vocabulary to below 64k words based on occurrence count. This limit was also applied in the Kaldi systems, a restriction we plan to remove in future.

We also investigated the use of RNN models, which were interpolated with the 4-gram model, and used to rescore the 3-gram lattices. However, we did not use these models in the system, as we were unable to observe any performance improvements over the large 4-gram model on its own. This is probably due to the fact that the RNNs available at the time of submission were trained on much smaller quantities of text.

Corpus	Quantity (hrs)
TED talks	143
Switchboard	285
AMI meetings (a)	127
AMI meetings (b)	78

Table 3: Training data quantities

3.2. Acoustic modelling

3.2.1. Training data

For in domain training data, as in previous years, we used 813 TED talks recorded prior to the end of 2010, which were aligned to the transcriptions available online using an efficient lightly-supervised technique [11]. We also used two sources of out-of-domain data: the Switchboard 1 corpus of conversational telephone speech, and the AMI corpus of multi-party meetings[1]. The quantities of speech data are summarised in Table 3.

As can be seen from the table, we use the AMI meetings corpus in two configurations. Previously, we have assumed that the AMI corpus is not well-matched to the TED domain, and used it purely as a means of generating bottleneck features for the MLAN scheme described in Section 3.2.2. In this case, we use a setup (a) described in [12]. Following last year's evaluation, however, we observed that with the passing of time, the changing format and expanding scope of TED talks has led to the pre-2010 data no longer being the best match for future test sets. This year, therefore, we decided to train one set of acoustic models on a combination of the TED and AMI data. In this case, we used a more recently-defined training setup (b) that aims to be reproducible by other sites and forms the basis of a Kaldi recipe. This is described in detail in [13].

3.2.2. Tandem MLAN systems

The multi-level adaptive networks (MLAN) scheme [14] aims to make optimal use of mismatched OOD data in training a system for which limited data is available for the target domain. Taking advantage of the fact that features derived from neural networks are known to be portable across domains, OOD DNNs with a bottleneck layer [15] are used to generate features for the in-domain data. In the MLAN scheme, a second-level network is trained on these features, augmented with the original acoustic features, to ensure robustness when the input bottleneck features are poorly-matched to the new domain, and – since each DNN incorporates several frames of acoustic context – allowing wider acoustic context to be incorporated without additional parameters.

The MLAN scheme has a particular advantage when used with the Switchboard telephone data, as it allows us to make good use of narrowband data without the need for upsampling, which may cause performance degradation. To do this, the first level nets are trained on the 8khz Switchboard data. To generate features for the TED data, we can simply downsample this data in to match the telephone data. The bottleneck features are then augmented with standard acoustic features derived *without* the need for any change in sample rate.

[1] http://corpus.amiproject.org/

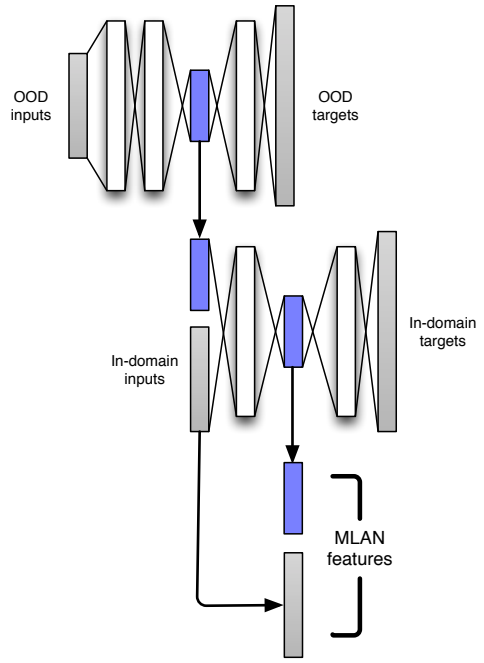


Figure 2: Tandem MLAN feature generation

In this year's system, we used MLAN purely in a tandem configuration [16], whereby the final bottleneck features are augmented with the original acoustic features and used to train a GMM. The complete feature generation process is illustrated in Figure 2. The advantage of this configuration is that it allows us to take advantage of the large quantity of training data available for each test speaker in the TED task by estimating multiple CMLLR adaptation transforms per speaker with a regression class tree.

All tandem networks use 6 wide layers with 2048 hidden units per layer; the bottleneck layers have 30 units. The nets are trained with the standard cross-entropy criterion using approximately 6,000 context-dependent triphone targets derived from a baseline GMM. Input acoustic features are PLPs with first and second derivatives – 39 features in total. Both first- and second-level networks use 9 frames of acoustic context. The final GMMs have MPE training applied. All tandem systems use HTK, as we were unable to achieve comparable performance with Kaldi on these features.

3.2.3. Hybrid LHUC systems

We have previously used DNNs in a hybrid configuration, whereby the nets are used to generate posterior probabili-

ties over tied-state triphones for direct use in the decoder. We have noted that speaker adaptation, using a global fMLLR transform per speaker, is essential for competitive performance on the TED task. This year, in addition, we experimented with the use of our recently-proposed technique [5] for creating speaker-dependent DNNs by adapting each hidden layer on a per-speaker basis, which we term Learning Hidden Unit Contributions (LHUC). We briefly summarise it here. Consider the l-th hidden layer of a DNN, given by

$$\mathbf{h}^l = \phi^l \left(\mathbf{W}^{l\top} \mathbf{h}^{l-l} \right). \quad (2)$$

where $\mathbf{W}^{l\top}$ are the weights and ϕ^l is the nonlinear transfer function at the l-th hidden layer. We modify a standard speaker independent (SI) DNN by defining a set of speaker-dependent (SD) parameters for speaker m, $\theta^m = \{\mathbf{r}_m^1, \ldots, \mathbf{r}_m^L\}$, where $\mathbf{r}_m^l \in \mathbb{R}^{M^l}$ is the vector of SD parameters for the lth hidden layer. If $a(\mathbf{r}_m^l)$ is element-wise function that constrains the range of $\mathbf{r}_m^l$, then we can modify (2) to define a hidden layer output that is specific to speaker m:

$$\mathbf{h}_m^l = a(\mathbf{r}_m^l) \circ \phi^l \left(\mathbf{W}^{l\top} \mathbf{h}_m^{l-l} \right), \quad (3)$$

where $\circ$ is an element-wise multiplication. The SD term can be viewed as applying different weights to the contributions from each the hidden units on a per-speaker basis. We define $a(\cdot)$ as a sigmoid with amplitude 2, $a(c) = 2/(1+\exp(-c))$, so that each speaker-dependent weighting is strictly positive and centered at one. This re-parametrisation is for optimisation purposes only; at runtime $a(\cdot)$ can be evaluated once for a given set of θ^m and directly used as a scaling factor. The SD parameters are optimised with respect to the negative log posterior probability $\mathcal{F}(\theta^m)$ over T^m adaptation data-points of the m-th speaker, similar to the SI case:

$$\mathcal{F}(\theta^m) = -\sum_t^{T^m} \log P(s_t | \mathbf{x}_t^m; \theta^m). \quad (4)$$

given speech samples $\mathbf{x}_t$ and tied state labels s_t

We investigated the use of LHUC with three different non-linearities ϕ^l: in addition to the standard sigmoid, we use rectifying linear units [7] and Maxout units [17] which we proposed for ASR in [6]. Rather than applying any explicit function, the maxout network groups linear activations, and passes forward the maximum value in each group:

$$h_i^l = \max_{k=0}^{K-1} (z_{j+k}^l), \quad j = i.K \quad (5)$$

where the z_i^l are the linear outputs of the l-th layer.

Our hybrid DNNs again use 2048 hidden units per layer, but with 12,000 tied-state outputs. The input features are again PLPs with first and second derivatives, and 9 frames of context in total. For the maxout non-linearity we set the number of hidden maxout units to 1500, with a group size of two. All models had fMLLR applied to the input feature space. The LHUC nets were trained only on the 143 hours of TED data. All adaptation on the test set was performed on a per-talk basis using the output from a first-pass decode.

We also trained a single DNN system on a combination of the TED data and the AMI corpus setup (b), with sequence training following the recipe of [8]. As we will show in the results section, the use of the AMI corpus appears to particularly benefit performance on `tst2013`, perhaps due to its poorer match to the pre-2010 TED data.

3.3. Results

We present development results on `tst2011` generated with manual segmentations. Table 4 compares performance of tandem MLAN systems with a baseline trained purely on in-domain features. Consistent with previous results, it may be seen that the use of OOD data gives significant performance improvements: it is interesting to see that the use of entirely mismatched narrowband telephone speech from Switchboard still leads to a 13.5% relative WER reduction with the 3gram LM. The results of the Hybrid LHUC systems are shown in Table 5 (these results are not fully comparable with the results from the previous table as a weaker LM is used). The LHUC technique leads to gains with all three types of non-linearity investigated, and appears to be complementary to the use of fMLLR transforms on the input space. Both the ReLU and Maxout non-linearities appear to derive greater benefit from LHUC.

Model	3gram	4gram
Baseline tandem	12.6	-
SWB MLAN	10.9	10.3
AMI MLAN	11.2	9.8
ROVER	-	9.3

Table 4: Tandem MLAN DNN development results on `tst2011`. All systems are trained with MPE.

Model	WER (%)
DNN	15.2
+LHUC	13.7 (-9.9)
+fMLLR	13.9 (-8.5)
+LHUC	12.9 (-15.1)
ReLU	15.2
+LHUC	13.5 (-11.2)
+fMLLR	13.6 (-10.5)
+LHUC	12.7 (-16.4)
Maxout	14.3
+LHUC	12.8 (-10.4)
+fMLLR	12.5 (-12.6)
+LHUC	11.9 (-16.8)

Table 5: Hybrid DNN development results on `tst2011` using weak 3gram LM. Relative improvements are given in parentheses w.r.t. the corresponding SI model.

Model	WER (%)
2013 systems	
AMI MLAN	22.9
Final submission	21.5
HTK tandem systems	
AMI MLAN	18.1
SWB MLAN	17.2
Kaldi hybrid systems	
ReLU + LHUC	18.4
MaxOut + LHUC	18.7
TED+AMI Seq	15.3
ROVER combinations	
Tandem MLAN	16.6
All Hybrid	15.3
All systems	**14.4**

Table 6: Final systems with automatic segmentation on `tst2013`

Finally, we present results on `tst2013` with automatic segmentation in Table 6. All these results use lattice rescoring with the 751MW 4gram model. The system comination weights for ROVER were tuned on the development sets `dev2010`, `tst2010` and `tst2011`. Note that our scoring is not entirely consistent with that performed in the 2013 evaluation: we obtain hypothesis-to-reference alignments over the entire talk, rather than on a per utterance basis. We believe this approach is fairer as it makes the scoring more robust to slight discrepancies in segment timings between the human reference and the automatic system, which can lead to single words being counted as a deletion error in one segment and an insertion error in the adjoining segment. For comparison, our final 2013 scores 21.5% with full-talk scoring, compared to 22.1% by the official method.

From the table, we see that the new VAD system gives an absolute WER reduction of 3.8% on the AMI MLAN system, which is otherwise unchanged from 2013. Again, the two tandem MLAN systems are highly complementary when used in combination; the sequence-trained DNN trained with both TED and AMI data seems to perform particularly well on the `tst2013`, perhaps reflecting the more diverse range of accents in this test set. Finally, the tandem and hybrid systems are seen to be complementary, resulting in a further reduction in WER to 14.4%. On the `tst2014` test set, this final system has an official score of 12.7%. However, as noted above, this result includes a number of erroneous insertions and deletions at utterance boundaries. Scoring on a per-talk basis against the same reference transcription yields a WER of 10.7%.

4. German system

A major hurdle in achieving high-quality recognition lies in the collection of appropriate training data, both for acoustic modelling and language modelling. For acoustic modelling, participants in this year's ASR evaluation track were provided with German data from the Euronews corpus, a speech corpus that contains news broadcasts in a multitude of languages [18, 19]. The permitted training data was not limited to Euronews, however. Any speech recording made before a certain cut-off date (17/07/2012) could be included. We have chosen to include recordings of plenary sessions of the European parliament, made between January 2007 and December 2010. These recordings are publicly available online, along with their approximate transcriptions [20, 21]. Both text and audio are available in German making this data readily usable for acoustic model training. We will henceforth refer to this set of data as *Europarl*. Lastly, we have included the GlobalPhone corpus in the training data [22].

For LM training, we used the same method that was described in [20] and used in the ASR track of IWSLT 2013. Briefly, it consists of selecting 30% of the training data according to maximum cross-entropy with the target domain [23]. Then, a 3-gram language model is trained on this selected data using Kneser-Ney smoothing and a vocabulary is determined by selecting the top 1-grams in this model, ranked according to decreasing smoothed 1-gram probability. Finally, 4-gram LM training is performed on the same data selection, in which the words are restricted to those in the chosen vocabulary. RNN language model were trained using the RNNLM toolkit [24]. During evaluation, these RNN models were used to rescore 100-best lists, i.e. the 100 most likely utterance recognition hypotheses, that were generated with the 4-gram LM.

4.1. Language Modelling

German Language models were trained on all the German monolingual text corpora provided in the ACL statistical machine translation workshop 2014 [25], and the in-domain text data provided by the organisers of IWSLT 2014. They are listed in table 7. The text in each of these corpora was tokenised as follows: first, all the punctuation is removed. Then all numbers in the text are expanded, as are the most common units, e.g. currency, distance, volume, weight, etc. Any word that is completely capitalised, or in which the letters are separated by full stops, is treated as an abbreviation, and its letters are spelled out. For further details, see [20].

Full-sized 4-gram LMs are trained on each of these text corpora, after which they are interpolated. The interpolation weights are optimised, so as to reduce the perplexity of the resulting LM on an in-domain text corpus, here the text of `dev2012`. Since the list of words contained in this LM is prohibitively large for ASR, it has to be limited to the top words in the ranked list described above. Choosing the size of the vocabulary is a trade-off between model perplexity and OOV-rate, as is shown in Table 8. We have opted for a vocabulary of size 300k. This list of words is turned into a lexicon for ASR, as discussed below, in section 4.2. We will refer to this lexicon as $dict_1$. Since the final 4-gram LM is too large to use in ASR directly, we prune it with a threshold of

corpus	10^6 words
Europarl-v7	47.4
News Commentary	4.5
News Crawl 2007	31.5
News Crawl 2008	107.9
News Crawl 2009	101.6
News Crawl 2010	45.9
News Crawl 2011	252.8
News Crawl 2012	319.7
News Crawl 2013	543.0
IWSLT	2.8
Total	1455.0

Table 7: The different training corpora used for German language modelling, and their sizes

#words	ppl	oov-rate (%)
$1 \cdot 10^5$	235.45	4.22
$2 \cdot 10^5$	261.49	2.85
$3 \cdot 10^5$	274.33	2.36
$4 \cdot 10^5$	280.29	2.14

Table 8: Perplexities on `dev2012`, along with the OOV-rate of the resulting 4-gram LMs, limited to different vocabulary sizes.

10^{-7}. The resulting reduced LM will be referred to below as LM_1. For RNN training, the vocabulary was further reduced to 50k, for computational reasons. We train it on a random selection of 10M lines from the corpora listed in table 8. The hidden layer of the network contains 30 nodes.

4.2. Acoustic Modelling

As discussed above, data sources available for German acoustic model training are Euronews, GlobalPhone, and Europarl. Since Europarl has only approximate transcriptions, we have to apply some form of light supervision on it, in order to obtain a subset in which the transcriptions are accurate. We do this using the same method as in [26]. We use an initial acoustic model, GMM_0, and a biased language model, LM_0, to perform recognition on the entire data, and define a new training set which contains only the segments where the recognition matches the approximate transcriptions. Although a new model trained on this set can in principle be used to repeat the procedure iteratively, there are no guarantees that models from such subsequent iterations will be significantly superior. On the contrary, one even runs the risk of degrading the model by applying this technique iteratively [27]. We have therefore only run a single iteration of data selection on Europarl. The biased Language model, LM_0, was obtained by interpolating the LM provided with GlobalPhone with a language model trained on the annotations of the Europarl speech data. The initial acoustic model, GMM_0, was trained on a combination of Euronews and GlobalPhone. The acoustic features were extracted in frames of 25 ms, with a shift 10 ms. 13 MFCC coefficients in each frame were stacked within context windows of 9 frames, and the resulting 117-dimensional representations were projected down to 40 dimensions using LDA/MLLT [28]. GMM_0 has 3000 context dependent states, with a total of 48000 Gaussians. No adaptation was performed. From an initial estimated total of 733 hours of Europarl data, this model allows us to select about 80 hours. This number may seem small, but the total data is likely an overestimate due to overlapping speech segments. Moreover, the majority of the data consists of non-German segments, the speech and transcriptions of which are translated into German separately. The disagreement between text and audio is therefore very large. The amount of useful data from each corpus is listed in table 9, where GP stands for GlobalPhone, EN for Euronews, and EP for Europarl.

corpus	GP	EN	EP	total
#hours	14.85	57.35	79.90	152.10

Table 9: The size of all the different data sources for acoustic model training.

data	WER (%)
GP	49.64
+ EN	44.05
+ EP	41.38

Table 10: Word Error Rates on `dev2012` using different acoustic models

To demonstrate the benefits of adding each of these data sets, we have trained simple acoustic models on GlobalPhone (GP), on a combination of GlobalPhone and Euronews (GP+EN), and on all data combined (GP+EN+EP). The dictionary used in this training, which we will call $dict_0$ comprises the GlobalPhone dictionary, augmented with all the OOV words from the three training sets, altogether about 140000 words. The transcription of new words is generated with Sequitur G2P [29], trained on the 40000 words of GlobalPhone. The performance of the resulting models was evaluated on `dev2012`. The WERs are shown in table 10. We can see that, even though the domains of the different training sets are quite far apart, and none close to that of the development set, they all contribute to some extent in improving the results. We will therefore use a combination of these three sets for all acoustic model training that follows. The error rates shown in table 10 are rather high because little effort was taken to tune these evaluations to the target domain. $dict_0$ is a relatively small dictionary (for German), and the language model LM_0 is biased towards Europarl, not TED.

Using all available training data, i.e. GP+EP+EN, we perform speaker-adaptive training in order to obtain speaker dependent GMM-HMM models. The number of context dependent states in this new model was set at 9000, and

the number of Gaussians to 100000. We call this model GMM_1. Repeating the evaluation above with this model yields a WER of 37.65%, an absolute improvement of almost 4%. When we use the same acoustic model in conjunction with the LM_1, the pruned LM trained in section 4.1 and its associated dictionary, $dict_1$, the WER decreases further to 35.88%. This improvement is quite modest, considering the complexity of this LM and the fact that is specifically optimised for the TED domain. A likely reason is that the dictionary only contains about 40000 pronunciations that were manually transcribed. All the others have been generated using a grapheme-to-phoneme (G2P) conversion. All errors made during this process are propagated further through the ASR evaluation. To reduce this problem, we have performed dictionary expansion as proposed in [9]. In practice, we used G2P to generate the 10 most likely pronunciations for every word in dictionary $dict_1$, including the 40000 from the original GlobalPhone lexicon. For the latter, if none of the 10 generated pronunciations matched the original phonetic transcription, it was added as an 11th pronunciation. Initially, all pronunciations of a word are assigned a uniform probability. An alignment of the training data using model GMM_1 is then made, where the different pronunciations of each word of the transcription are set in parallel. The resulting alignments show the pronunciation of each word that best fits its acoustic realisation. Counting the occurrences of each pronunciation then allows an update of their probability in the dictionary, and a re-alignment. This is an iterative process in which the dictionary is refined in each iteration. Every few iterations, the acoustic model can be retrained as well. Here, we have chosen to do just 2 iterations, in each of which the acoustic model is retrained. We will refer to the resulting acoustic model as GMM_2. The resulting dictionary, $dict_2$, is an improvement over $dict_1$, not only because it contains pronunciation probabilities, but also because it lists pronunciations that make sense acoustically, rather than enforcing G2P's best guess. We ran an evaluation on `dev2012` with this pronunciation lexicon, using GMM_1 and the pruned LM, LM_1. The resulting WER was 29.86%, an absolute improvement of almost 6% compared to the original $dict_1$. When replacing the acoustic model GMM_1 for GMM_2, the WER becomes slightly higher: 30.91%. A possible explanation is that the degrees of freedom introduced by pronunciation variation allow the model to over-train.

A DNN is then trained up in a hybrid configuration with model GMM_2. This DNN consists of 6 hidden layers, with 2048 nodes each, connecting through a logistic sigmoid non-linearity. The output layer performs a softmax operation. At the input of the network are the MLLT-transformed speaker-adapted MFCC features we described above, stacked within a context window of 11 frames, which results in a 440-dimensional representation per frame (40$\times$ 11). The output is a vector of posterior probabilities over the context-dependent states of the GMM, converted into scaled likelihoods using prior probabilities obtained from training data [30]. The network is pre-trained with layer-wise RBM training, and finetuned by optimising a negative log-likelihood cost function. Evaluating this hybrid DNN setup on `dev2012` gives a WER of 27.83%. Note that all results thus far have either been obtained with the Europarl LM, or with a heavily pruned LM optimised for TED. The full TED-specific model has not been used due to computational limitations. We can, however, rescore the results with this larger LM, obtaining further reductions in WER. Similarly, all of the previous results can be rescored using the RNNLM. All results on `dev2012` are summarised in table 11.The system has an official score of 35.7% on the `tst2014` test set.

GMM_1 + $dict_0$ + LM_0	37.65
GMM_1 + $dict_1$ + LM_1	35.88
GMM_1 + $dict_2$ + LM_1	29.86
GMM_2 + $dict_2$ + LM_1	30.91
+ LM rescore	28.07
+ RNNLM rescore	27.59
GMM_2 + DNN + $dict_2$ + LM_1	27.83
+ LM rescore	25.33
+ RNNLM rescore	**24.90**

Table 11: The results of the German system on `dev2012`

5. Conclusions

We have described our ASR systems for the English and German 2014 IWSLT evaluation. Improvements to our English system, most particularly the use of AMI data, and the deployment of hybrid DNNs with LHUC and sequence training, result in a relative WER reduction of around 30% on the challenging `tst2013` evaluation set compared to our 2013 system. We intend to carry over these benefits to our German system, where a lack of suitable training data remains a challenge.

In the future, we plan to further investigate methods for robust DNN training and adaptation when the training data is limited or poorly-transcribed, something which should enable us to develop systems in new languages more rapidly. We also plan to work on removing the dependence on a dictionary completely, perhaps by adapting grapheme-based models. We also aim to re-incorporate RNN language models in our most competitive English system.

6. References

[1] E. Hasler, P. Bell, A. Ghoshal, B. Haddow, P. Koehn, F. McInnes, S. Renals, and P. Swietojanski, "The UEDIN systems for the IWSLT 2012 evaluation," in *Proc. IWSLT*, 2012.

[2] P. Bell, M. J. F. Gales, P. Lanchantin, X. Liu, Y. Long, S. Renals, P. Swietojanski, and P. Woodland, "Transcription of multi-genre media archives using out-of-

domain data," in *Proc. IEEE Workshop on Spoken Language Technology*, Dec. 2012.

[3] P. Bell, F. McInnes, S. R. Gangireddy, M. Sinclair, A. Birch, and S. Renals, "The UEDIN english ASR system for the IWSLT 2013 evaluation," in *Proc. IWSLT*, 2013.

[4] H. Yamamoto, Y. Wu, C. Huang, X. Lu, P. Dixon, S. Matsuda, C. Hori, and H. Kashioka, "The NICT ASR system for IWSLT 2012," in *Proc. International Workshop on Spoken Language Translation*, Hong Kong, Dec. 2012.

[5] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. SLT*, Lake Tahoe, USA, December 2014.

[6] P. Swietojanski, J. Li, and J.-T. Huang, "Investigation of maxout networks for speech recognition," in *Proc ICASSP*, 2014.

[7] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010.

[8] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, Lyon, France, August 2013.

[9] L. Lu, A. Ghoshal, and S. Renals, "Acoustic data-driven pronunciation lexicon for large vocabulary speech recognition," in *Proc. ASRU*, 2013.

[10] M. Sinclair, P. Bell, A. Birch, and F. McInnes, "A semi-markov model for speech segmentation with an utterance-break prior," in *Proc. Interspeech*, 2014.

[11] A. Stan, P. Bell, and S. King, "A grapheme-based method for automatic alignment of speech and text data," in *Proc. SLT*, Miama, Florida, USA, Dec. 2012.

[12] T. Hain, L. Burget, J. Dines, P. Garner, F. Grezl, A. Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "Transcribing meetings with the AMIDA systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 486–498, 2012.

[13] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *Proc. ASRU*, 2013.

[14] P. Bell, P. Swietojanski, and S. Renals, "Multi-level adaptive networks in tandem and hybrid ASR systems," in *Proc. ICASSP*, 2013.

[15] F. Grézl, M. Karafiát, S. Kontar, and J. Černokcý, "Probabilistic and bottleneck features for LVCSR of meetings," in *Proc. ICASSP*, 2007.

[16] H. Hermanksy, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000, pp. 1635–1630.

[17] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," *arXiv:1302.4389*, 2013.

[18] R. Gretter, "Euronews: A multilingual speech corpus for ASR," in *Proc LREC*, Reykjavik, Iceland, May 2014.

[19] ——, "Euronews: A multilingual benchmark for ASR and LID," in *Proc Interspeech*, Singapore, September 2014.

[20] J. Driesen, P. Bell, M. Sinclair, and S. Renals, "Description of the UEDIN system for German ASR," in *Proc IWSLT*, Heidelberg, Germany, December 2013.

[21] "The website of the european parliament." [Online]. Available: http://europarl.europa.eu

[22] T. Schultz, "GlobalPhone: A multilingual speech and text database developed at karlsruhe university," in *Proc. Interspeech*, Denver, Colorado, USA, 2002.

[23] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proc. ACL*, Uppsala, Sweden, July 2010.

[24] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, Makuhari, Japan, September 2010.

[25] "The website of the ACL statistical machine translation workshop," 2014. [Online]. Available: www.statmt.org/wmt14

[26] J. Driesen and S. Renals, "Lightly supervised automatic subtitling of weather forecasts," in *Proc. Automatic Speech Recognition and Understanding Workshop*, Olomouc, Czech Republic, December 2013.

[27] R. Zhang and A. Rudnickey, "A new data selection approach for semi-supervised acoustic modelling," in *Proc ICASSP*, Toulouse, France, May 2006.

[28] M. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[29] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, May 2008.

[30] H. Bourlard and N. Morgan. Kluwer Academic Publishers, 1994.

Improving MEANT Based Semantically Tuned SMT

Meriem Beloucif, Chi-kiu Lo, Dekai Wu

HKUST
Human Language Technology Center
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
{mbeloucif|jackielo|dekai}@cs.ust.hk

Abstract

We discuss various improvements to our MEANT tuned system, previously presented at IWSLT 2013. In our 2014 system, we incorporate this year's improved version of MEANT, improved Chinese word segmentation, Chinese named entity recognition and dedicated proper name translation, and number expression handling. This results in a significant performance jump compared to last year's system. We also ran preliminary experiments on tuning to IMEANT, our new ITG based variant of MEANT. The performance of tuning to IMEANT is comparable to tuning on MEANT (differences are statistically insignificant). We are presently investigating if tuning on IMEANT can produce even better results, since IMEANT was actually shown to correlate with human adequacy judgment more closely than MEANT. Finally, we ran experiments applying our new architectural improvements to a contrastive system tuned to BLEU. We observed a slightly higher jump in comparison to last year, possibly due to mismatches of MEANT's similarity models to our new entity handling.

1. Introduction

In this paper we present an improved version of our MT system tuned against MEANT (Lo and Wu [1, 2]; Lo *et al*. [3]), a semantic MT evaluation metric which has been proven to highly correlate with human adequacy judgments. We employ an improved version of MEANT that correlates more closely with human adequacy judgments, resulting also in translation performance gains compared to the system tuned against our previous version of MEANT from the IWSLT 2013 evaluation campaign (Lo *et al.* [4]). This improved variant of MEANT uses f-score to aggregate lexical similarities within role filler phrases instead of linear average. We also introduced several changes to last year's baseline, including improved Chinese word segmentation, improved Chinese named entity recognition combined with dedicated proper name translation, and number expression handling.

We also experimented with tuning against IMEANT (Wu *et al.* [5]), a new inversion transduction grammar (ITG) version of MEANT, that was shown this year to correlate with human adequacy judgements more closely than MEANT. Despite this fact, we observed that tuning to IMEANT is statistically indistinguishable from tuning to MEANT.In the past few years, MT research has mainly focused on evaluation using fast and cheap n-gram based MT evaluation metrics such as BLEU [6] which assume that a good translation is one that has similar lexical n-grams as the reference translation. Although such metrics tend to enforce fluency, it has been shown that these metrics generally do not emphasize meaning preservation, and thus are weak at enforcing translation adequacy (Callison-Burch *et al.* [7]; Koehn and Monz [8]).

Unlike BLEU, or other n-gram based metrics, the MEANT family of metrics adopt the principle that a good translation is one in which humans can successfully understand the central meaning of the input sentence as captured by the basic event structure "*who did what to whom, when, where and why"* (Pradhan *et al.* [9]). MEANT measures similarity between an MT output and a reference translation by comparing the similarities between the semantic frame structures of the MT output and reference. We have shown that MEANT correlates better with human adequacy judgments than commonly used MT evaluation metrics such as BLEU [6], NIST [10], METEOR [11], CDER [12], WER [13], and TER [14].

2. Related work

Surface-form oriented metrics like BLEU [6], NIST [10], METEOR [11], CDER [12], WER [13], and TER [14] do not correctly reflect the meaning similarities of the basic event structure "*who did what to whom, when, where and why*" of the input sentence. In fact, many studies (Callison-Bursh *et al.* [7]; Koehn and Monz [8]) report cases where BLEU strongly disagrees with human adequacy judgment. This has caused a recent surge of work on developing MT evaluation metrics that outperforms BLEU in correlation with human judgment. AMBER [15] shows a high correlation with human adequacy judgment (Callison-Burch *et al.* [16]); however, it is very hard to indicate what errors the MT systems are making.

Many automatic metrics that aggregate semantic similarity have been introduced, but no tuning has been done using these metrics, because of their expensive run time. Gimenez and Marquez [17, 18] introduced ULC, an automatic metric that incorporates several semantic similarity features and shows improved correlation with human judgement of translation quality [19, 17, 20, 18]. SPEDE [21] is a metric that integrats probabilistic FSM and PDA models that predicts the edit sequence needed for the MT output to match the reference. SAGAN [22] is a semantic textual similarity metric based on a complex textual entailment pipeline. These aggregated metrics require sophisticated feature extraction steps; furthermore, they typically rely on several dozens of parameters to tune and use expensive linguistic resources, like WordNet and paraphrase tables. These metrics themselves are expensive in training and tuning due to the large number of parameters that need to be estimated, thus to tune against these metrics can be extremely expensive.

3. The MEANT family of metrics

3.1. MEANT

MEANT (Lo *et al.* [3]) is a weighted f-score over the matched semantic role labels of automatically aligned semantic frames and role fillers. MEANT outperforms BLEU, NIST, METEOR, WER, CDER and TER in correlation with human adequacy judgment. MEANT is easily portable to other languages requiring only an automatic semantic parser and a large monolingual corpus in the output language for identifying the semantic structures and to establish the lexical similarity between the semantic role fillers of the reference and translation.

More precisely, MEANT is computed as follows:

1. Apply an automatic shallow semantic parser to both the reference and machine translations. (Figure 1 shows examples of automatic shallow semantic parses on both reference and machine translations.)

2. Apply the maximum weighted bipartite matching algorithm to align the semantic frames between the reference and machine translations according to the lexical similarities of the predicates. ([23] proposed a backoff algorithm that evaluates the entire sentence of the MT output using the lexical similarity based on the context vector model, if the automatic shallow semantic parser fails to parse the reference or machine translations.)

3. For each pair of the aligned frames, apply the maximum weighted bipartite matching algorithm to align the arguments between the reference and machine translations according to the lexical similarity of role fillers.

4. Compute the weighted f-score over the matching role labels of these aligned predicates and role fillers as follow :

$$
\begin{aligned}
q^0_{i,j} &\equiv \text{ARG } j \text{ of aligned frame } i \text{ inMT} \\
q^1_{i,j} &\equiv \text{ARG } j \text{ of aligned frame } i \text{ in REF} \\
w^0_i &\equiv \frac{\text{\#tokens filled in aligned frame } i \text{ of MT}}{\text{total \#tokens in MT}} \\
w^1_i &\equiv \frac{\text{\#tokens filled in aligned frame } i \text{ of REF}}{\text{total \#tokens in REF}} \\
w_{\text{pred}} &\equiv \text{weight of similarity of predicates} \\
w_j &\equiv \text{weight of similarity of ARG } j \\
\mathbf{e}_{i,\text{pred}} &\equiv \text{the pred string of the aligned frame } i \text{ of MT} \\
\mathbf{f}_{i,\text{pred}} &\equiv \text{the pred string of the alignedframe } i \text{ of REF} \\
\mathbf{e}_{i,j} &\equiv \text{the role fillers of ARG } j \text{ of the aligned frame } i \\
\mathbf{f}_{i,j} &\equiv \text{the role fillers of ARG } j \text{ of the aligned frame } i \\
s(e,f) &= \text{lexical similarity of token } e \text{ and } f
\end{aligned}
$$

$$
\begin{aligned}
\text{prec}_{\mathbf{e},\mathbf{f}} &= \frac{\sum_{e\in\mathbf{e}} \max_{f\in\mathbf{f}} s(e,f)}{|\,\mathbf{e}\,|} \\
\text{rec}_{\mathbf{e},\mathbf{f}} &= \frac{\sum_{f\in\mathbf{f}} \max_{e\in\mathbf{e}} s(e,f)}{|\,\mathbf{f}\,|} \\
s_{i,\text{pred}} &= \frac{2\cdot \text{prec}_{\mathbf{e}_{i,\text{pred}},\mathbf{f}_{i,\text{pred}}} \cdot \text{rec}_{\mathbf{e}_{i,\text{pred}},\mathbf{f}_{i,\text{pred}}}}{\text{prec}_{\mathbf{e}_{i,\text{pred}},\mathbf{f}_{i,\text{pred}}} + \text{rec}_{\mathbf{e}_{i,\text{pred}},\mathbf{f}_{i,\text{pred}}}} \\
s_{i,j} &= \frac{2\cdot \text{prec}_{\mathbf{e}_{i,j},\mathbf{f}_{i,j}} \cdot \text{rec}_{\mathbf{e}_{i,j},\mathbf{f}_{i,j}}}{\text{prec}_{\mathbf{e}_{i,j},\mathbf{f}_{i,j}} + \text{rec}_{\mathbf{e}_{i,j},\mathbf{f}_{i,j}}}
\end{aligned}
$$

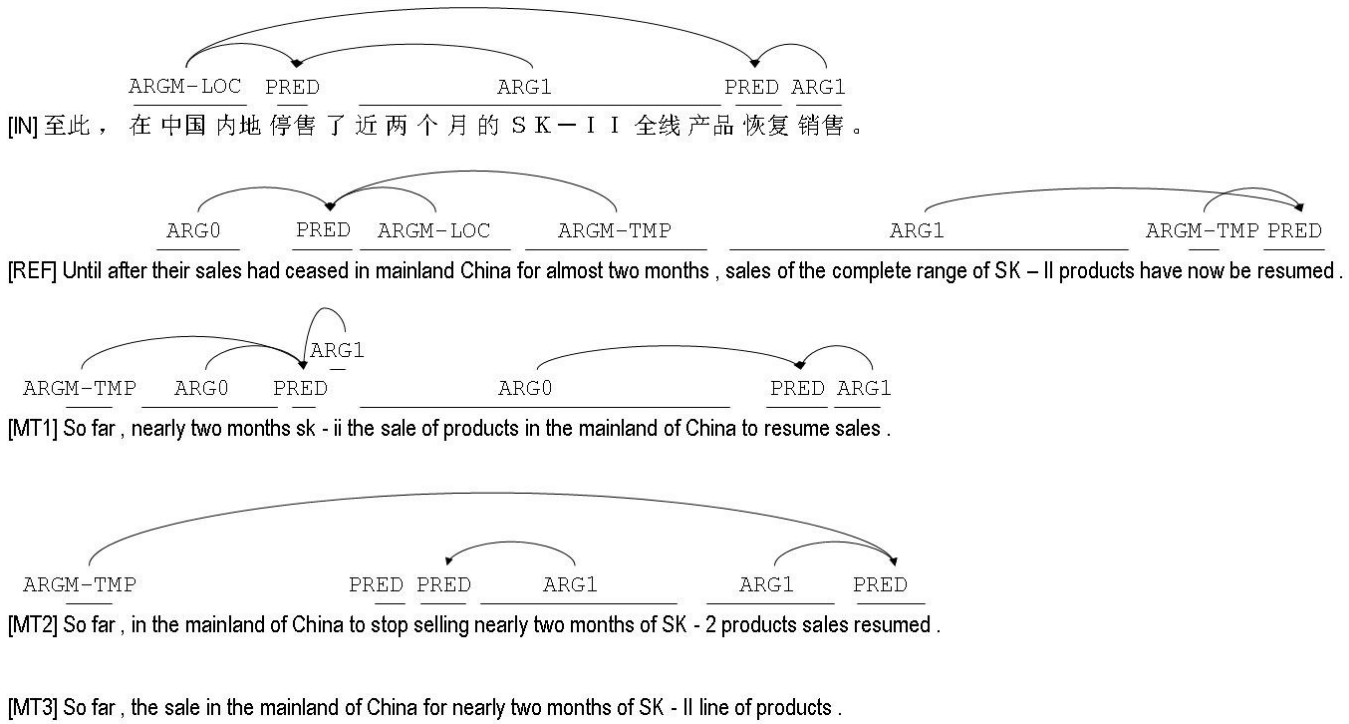


Figure 1: Examples of automatic shallow semantic parses. The input is parsed by a Chinese automatic shallow semantic parser. The reference and MT output are parsed by an English automatic shallow semantic parser. There are no semantic frames for MT3 since there is no predicate.

$$\begin{aligned}
\text{precision} &= \frac{\sum_i w_i^0 \frac{w_{\text{pred}} s_{i,\text{pred}} + \sum_j w_j s_{i,j}}{w_{\text{pred}} + \sum_j w_j |q_{i,j}^0|}}{\sum_i w_i^0} \\
\text{recall} &= \frac{\sum_i w_i^1 \frac{w_{\text{pred}} s_{i,\text{pred}} + \sum_j w_j s_{i,j}}{w_{\text{pred}} + \sum_j w_j |q_{i,j}^1|}}{\sum_i w_i^1} \\
\text{MEANT} &= \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\end{aligned}$$

where $q_{i,j}^0$ and $q_{i,j}^1$ are the argument of type j in frame i in MT and REF respectively. w_i^0 and w_i^1 are the weights for frame i in MT/REF respectively. These weights estimate the degree of contribution of each frame to the overall meaning of the sentence. w_{pred} and w_j are the weights of the lexical similarities of the predicates and role fillers of the arguments of type j of all frame between the reference translations and the machine translations. There is a total of 12 weights for the set of semantic role labels in MEANT as defined in Lo and Wu [24]. For MEANT, they are determined using supervised estimation via a simple grid search to optimize the correlation with human adequacy judgments (Lo and Wu [1]). For UMEANT (Lo and Wu [2]), they are estimated in an unsupervised manner using relative frequency of each semantic role label in the references and thus UMEANT is useful when human judgments on adequacy of the development set are unavailable.

3.2. IMEANT

IMEANT (Wu *et al.* [5]) is an inversion transduction grammar based variant of MEANT. IMEANT uses a a length-normalized weighted BITG [25, 26, 27, 28] to constrain permissible token alignment patterns between aligned role filler phrases. More precisely, IMEANT differs from MEANT in the definition of $s_{i,\text{pred}}$ and $s_{i,j}$, as follows:

$$\begin{aligned}
G &\equiv \langle \{\mathrm{A}\}, \mathcal{W}^0, \mathcal{W}^1, \mathcal{R}, \mathrm{A} \rangle \\
\mathcal{R} &\equiv \{\mathrm{A} \rightarrow [\mathrm{AA}], \mathrm{A} \rightarrow \langle \mathrm{AA} \rangle, \mathrm{A} \rightarrow e/f\}
\end{aligned}$$

$$\begin{aligned}
p([\mathrm{AA}]|\mathrm{A}) &= p(\langle \mathrm{AA} \rangle | \mathrm{A}) = 1 \\
p(e/f|\mathrm{A}) &= s(e,f) \\
s_{i,\text{pred}} &= \lg^{-1}\left(\frac{\lg\left(P\left(\mathrm{A} \overset{*}{\Rightarrow} \mathbf{e}_{i,\text{pred}}/\mathbf{f}_{i,\text{pred}}|G\right)\right)}{\max(|\mathbf{e}_{i,\text{pred}}|, |\mathbf{f}_{i,\text{pred}}|)}\right) \\
s_{i,j} &= \lg^{-1}\left(\frac{\lg\left(P\left(\mathrm{A} \overset{*}{\Rightarrow} \mathbf{e}_{i,j}/\mathbf{f}_{i,j}|G\right)\right)}{\max(|\mathbf{e}_{i,j}|, |\mathbf{f}_{i,j}|)}\right)
\end{aligned}$$

where G is a bracketing ITG whose only non terminal is A, and $\mathcal{R}$ is a set of transduction rules with $e \in \mathcal{W}^0 \cup \{\epsilon\}$ denoting a token in the MT output (or the *null* token) and $f \in \mathcal{W}^1 \cup \{\epsilon\}$ denoting a token in the reference translation (or the *null* token).

The rule weight function p is set to be 1 for structural transduction rules, and for lexical transduction rules it is defined using MEANT's context vector model based lexical similarity measure. The Saers *et al.* [29] algorithm is used to compute the inside probability of a pair of segments, $P\left(\mathrm{A} \stackrel{*}{\Rightarrow} \mathbf{e}/\mathbf{f}|G\right)$.

Given this, $s_{i,\text{pred}}$ and $s_{i,j}$ now represent the length normalized BITG parse scores of the predicates and role fillers of the arguments of type j between the reference and machine translations.

4. Baseline

In this section, we describe in detail our systems for the Chinese-English and English-Chinese TED talk MT tasks in terms of data, preprocessing, SMT pipeline and MEANT settings.

4.1. Data

Our main goal for 2014 was to improve our MEANT tuned system and compare the results to our 2013 system. For this purpose, we deliberately constrained our training data to 2013 in-domain data only. Thus we use the English-Chinese parallel data from the IWSLT 2013 training set and used the output side to train the language model.

Similarly, our development set was restricted to the IWSLT 2013 development set. Since our main focus was to test our performance in comparison to 2013, we purposely targeted the IWSLT 2013 set more than the IWSLT 2014 set. However, we do present IWSLT 2014 results for our BLEU tuned system for both English-Chinese and Chinese-English.

The English sentences were normalized for punctuation, tokenization, and truecasing.

Obviously, higher scores could have been obtained by training on the IWSLT 2014 data set instead of 2013.

4.2. SMT pipeline

With the goal of improving MT utility by using our new improved version of MEANT as an objective function to drive minimum error rate training (MERT) [30] of state-of-the-art MT systems, we set up our baseline using the translation toolkit Moses [31]. In our experiments, we are using the flat phrase-based MT. The language models are trained using the SRI language model toolkit [32]. For both translation tasks, we used a 6-gram language model. We use ZMERT [33] to tune the baseline since it is a reliable implementation of MERT and is fully configurable and extensible allowing us to easily incorporate our new evaluation metrics.

5. Experiments

5.1. MEANT improvements

This year's system incorporated new improvements to the MEANT metric, consisting of using f-score in order to aggregate lexical similarities *within* semantic role filler phrases instead of Mihalcea's [34] method used in our last year system. We also tried to extend the window-size from 3 to 5 for the context vector model trained on the word segmented monolingual English gigaword corpus.

Since UMEANT (Lo and Wu [35]) has been shown to be more stable when evaluating translations across different language pairs (Machacek and Bojar [36]), we use UMEANT for evaluating our output.

5.2. Tuning to IMEANT

In this paper, we also ran preliminary experiments on tuning to IMEANT [5], the new inversion transduction grammar based variant of MEANT, that achieves higher correlation with human adequacy judgments of MT output quality than MEANT and its variants. Addanki *et al.* [28] showed empirically that the semantic role reordering that MEANT uses is covered by ITG constraints.

5.3. Word segmentation improvements

For Chinese sentences, we improved the segmentation of Chinese words. We performed extensive comparisons between four word segmentation approaches. The results reported this year were obtained using the ICTCLAS word segmenter [37].

5.4. Named entity translation improvements

We also used our own new implementation of Chinese named entity recognition and a dedicated proper name translation, where we use our own library translator based on Wikipedia data. We implemented an adequate library generator for our new named entity recognizer.

Table 1: Translation quality of the participated Chinese-English MT systems on the IWSLT 2013 test set: (a) 2013 MEANT-tuned system, (b) 2014 improved MEANT-tuned system.

	uncased (internal)							
System	BLEU	NIST	METEOR	TER	WER	PER	CDER	MEANT
2013 MEANT-tuned system	10.49	4.54	4.24	73.97	75.77	59.17	70.94	31.42
2014 MEANT-tuned system	**13.56**	**4.97**	**4.69**	**70.48**	**73.98**	**56.19**	**69.18**	**39.79**

Table 2: Translation quality of the participated Chinese-English MT systems on the IWSLT 2013 test set tuned against MEANT and IMEANT respectively.

	uncased (internal)							
System	BLEU	NIST	METEOR	TER	WER	PER	CDER	MEANT
MEANT-tuned	**13.56**	4.97	**4.69**	70.48	73.98	56.19	69.18	**39.79**
IMEANT-tuned	13.55	**4.99**	4.68	70.48	**73.60**	**55.78**	**68.85**	34.21

5.5. Number expression translation improvements

We incorporated our HKUST number expression recognition and translation module this year.

6. Results

For IWSLT 2014 we submitted our new architecturally changed baseline for the BLEU tuned system for both, English-Chinese TED talks and Chinese-English TED talks as a primitive task. We also include our latest results on the MEANT-tuned Moses flat phrase-based system MT system, as well as our IMEANT-tuned system for Chinese-English TED talks MT task.

Table 1 shows that our new MEANT tuning using f-score as an aggregation function outperforms 2013 system. We see a high jump in terms of BLEU scores between all our MEANT tuned systems for last year and this year.

Table 2 shows also that IMEANT, the ITG variant of MEANT, produces almost identical results in comparison to our MEANT-tuned system. The differences are statistically insignificant. We are presently investigating whether tuning to IMEANT can produce even better results, since IMEANT was actually shown to correlate more closely with human adequacy judgment than MEANT.

Tables 3 and 4 show that our new word segmentation, named entity translation modules, and number expression translation modules incorporated in this year's system improved the performance of our BLEU and TER tuned systems respectively in comparison to our 2013 BLEU and TER tuned systems.

Tables 5 and 6 represent our official submitted systems for IWSLT 2014 evaluation campaign for Chinese-English and English-Chinese. We evaluate on both the 2013 and 2014 test sets. For English-Chinese translations, only the character level BLEU and TER were given.

7. Conclusion

In this paper we have presented an improved version of our MEANT tuned system which shows significant improvements over last year's model. The major changes to the system include improved Chinese word segmentation, improved Chinese named entity recognition, a new dedicated proper name translation and new number expression handling. We also experimented with tuning against IMEANT, our ITG based variant of MEANT. IMEANT performance was surprisingly similar to that of MEANT despite the fact that IMEANT has been shown to correlate better with human adequacy judgment than MEANT. We are currently looking at the possible reasons behind such a result.

8. Acknowledgment

This material is based upon work supported in part by the Hong Kong Research Grants Council (RGC) research grants GRF620811, GRF621008; and by the Defense Advanced Research Projects Agency (DARPA) under BOLT contract nos. HR0011-12-C-0014 and HR0011-12-C-0016, and GALE contract nos. HR0011-06-C-0022 and HR0011-06-C-0023; by the European Union under the FP7 grant agreement no. 287658; GRF612806.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the EU, DARPA or RGC.

Table 3: Translation quality of the participated Chinese-English MT systems on the IWSLT 2013 test set: (a) 2013 BLEU-tuned system, (b) 2014 improved BLEU-tuned system.

	uncased (internal)							
System	BLEU	NIST	METEOR	TER	WER	PER	CDER	MEANT
2013 BLEU-tuned system	11.16	4.61	4.32	74.69	77.17	59.15	71.84	31.46
2014 BLEU-tuned system	**13.85**	**5.01**	**4.55**	**68.70**	**72.27**	**54.91**	**67.45**	**32.93**

Table 4: Translation quality of the participated Chinese-English MT systems on the IWSLT 2013 test set: (a) 2013 TER-tuned system, (b) 2014 improved TER-tuned system.

	uncased (internal)							
System	BLEU	NIST	METEOR	TER	WER	PER	CDER	MEANT
2013 TER-tuned system	10.65	2.96	3.33	71.09	71.51	60.72	69.10	38.38
2014 TER-tuned system	**11.16**	**4.01**	**3.97**	**66.49**	**68.43**	**56.93**	**65.78**	**39.18**

9. References

[1] C.-k. Lo and D. Wu, "MEANT: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles," in *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011)*, 2011.

[2] ——, "Unsupervised vs. supervised weight estimation for semantic MT evaluation metrics," in *Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-6)*, 2012.

[3] C. Lo, A. K. Tumuluru, and D. Wu, "Fully automatic semantic MT evaluation," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012.

[4] C.-k. Lo, M. Beloucif, and D. Wu, "Improving machine translation into Chinese by tuning against Chinese MEANT," in *International Workshop on Spoken Language Translation (IWSLT 2013)*, 2013.

[5] D. Wu, C.-k. Lo, M. Beloucif, and M. Saers, "Better semantic frame based mt evaluation via inversion transduction grammars," 2014, sSST.

[6] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, Pennsylvania, July 2002, pp. 311–318.

[7] C. Callison-Burch, M. Osborne, and P. Koehn, "Re-evaluating the role of BLEU in machine translation research," in *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 2006.

[8] P. Koehn and C. Monz, "Manual and automatic evaluation of machine translation between european languages," in *Workshop on Statistical Machine Translation (WMT-06)*, 2006.

[9] S. Pradhan, W. Ward, K. Hacioglu, J. H. Martin, and D. Jurafsky, "Shallow semantic parsing using support vector machines," in *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*, 2004.

[10] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *The second international conference on Human Language Technology Research (HLT '02)*, San Diego, California, 2002.

[11] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, Michigan, June 2005. [Online]. Available: http://www.aclweb.org/anthology/W/W05/W05-0909

[12] G. Leusch, N. Ueffing, and H. Ney, "CDer: Efficient MT evaluation using block movements," in *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 2006.

[13] S. Nießen, F. J. Och, G. Leusch, and H. Ney, "A evaluation tool for machine translation: Fast evaluation for MT research," in *The Second International Conference on Language Resources and Evaluation (LREC 2000)*, 2000.

[14] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *7th Biennial Conference Association for Machine Translation in the Americas (AMTA 2006)*, Cambridge, Massachusetts, August 2006, pp. 223–231.

[15] B. Chen, R. Kuhn, and G. Foster, "Improving AMBER, an MT evaluation metric," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012, pp. 59–63.

Table 5: Translation quality of the submitted Chinese-English MT systems on : (a) IWSLT 14 test set, (b) IWSLT 13 test set.

System	BLEU-cased	TER-cased	BLEU-uncased	TER-uncased
2014 ZH-EN BLEU-tuned	09.64	76.66	10.83	74.15
2014 ZH-EN BLEU-tuned	11.89	72.32	13.08	70.09

Table 6: Translation quality of the submitted English-Chinese MT systems on : (a) IWSLT 14 test set, (b) IWSLT 13 test set.

System	Character BLEU	Character TER
EN-ZH BLEU-tuned	18.81	70.94
EN-ZH BLEU-tuned	16.41	74.34

[16] C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2012 workshop on statistical machine translation," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012, pp. 10–51.

[17] J. Giménez and L. Màrquez, "Linguistic features for automatic evaluation of heterogenous MT systems," in *Second Workshop on Statistical Machine Translation (WMT-07)*, Prague, Czech Republic, June 2007, pp. 256–264. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0738

[18] ——, "A smorgasbord of features for automatic MT evaluation," in *Third Workshop on Statistical Machine Translation (WMT-08)*, Columbus, Ohio, June 2008. [Online]. Available: http://www.aclweb.org/anthology/W/W08/W08-0332

[19] C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder, "(meta-) evaluation of machine translation," in *Second Workshop on Statistical Machine Translation (WMT-07)*, 2007.

[20] ——, "Further meta-evaluation of machine translation," in *Third Workshop on Statistical Machine Translation (WMT-08)*, 2008.

[21] M. Wang and C. D. Manning, "SPEDE: Probabilistic edit distance metrics for MT evaluation," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012.

[22] J. Castillo and P. Estrella, "Semantic textual similarity for MT evaluation," in *7th Workshop on Statistical Machine Translation (WMT 2012)*, 2012.

[23] C.-k. Lo and D. Wu, "Can informal genres be better translated by tuning on automatic semantic metrics?" in *14th Machine Translation Summit (MT Summit XIV)*, 2013.

[24] ——, "SMT vs. AI redux: How semantic frames evaluate MT more accurately," in *Twenty-second International Joint Conference on Artificial Intelligence (IJCAI-11)*, 2011.

[25] D. Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," *Computational Linguistics*, vol. 23, no. 3, pp. 377–403, 1997.

[26] R. Zens and H. Ney, "A comparative study on reordering constraints in statistical machine translation," in *41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, Stroudsburg, Pennsylvania, 2003, pp. 144–151. [Online]. Available: http://dx.doi.org/10.3115/1075096.1075115

[27] M. Saers and D. Wu, "Improving phrase-based translation via word alignments from stochastic inversion transduction grammars," in *Third Workshop on Syntax and Structure in Statistical Translation (SSST-3)*, Boulder, Colorado, June 2009, pp. 28–36. [Online]. Available: http://www.aclweb.org/anthology/W/W09/W09-2304

[28] K. Addanki, C.-k. Lo, M. Saers, and D. Wu, "LTG vs. ITG coverage of cross-lingual verb frame alternations," in *16th Annual Conference of the European Association for Machine Translation (EAMT-2012)*, Trento, Italy, May 2012.

[29] M. Saers, J. Nivre, and D. Wu, "Learning stochastic bracketing inversion transduction grammars with a cubic time biparsing algorithm," in *11th International Conference on Parsing Technologies (IWPT'09)*, Paris, France, October 2009, pp. 29–32.

[30] F. J. Och, "Minimum error rate training in statistical machine translation," in *41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, Sapporo, Japan, July 2003, pp. 160–167. [Online]. Available: http://www.aclweb.org/anthology/P03-1021

[31] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Interactive Poster and Demonstration Sessions of the 45th Annual Meeting of the Association for Computational Linguistics (ACL*

2007), Prague, Czech Republic, June 2007, pp. 177–180.

[32] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *7th International Conference on Spoken Language Processing (ICSLP2002 - INTERSPEECH 2002)*, Denver, Colorado, September 2002, pp. 901–904.

[33] O. F. Zaidan, "Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems," *The Prague Bulletin of Mathematical Linguistics*, vol. 91, pp. 79–88, 2009.

[34] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *The Twenty-first National Conference on Artificial Intelligence (AAAI-06)*, vol. 21, 2006.

[35] C.-k. Lo and D. Wu, "MEANT at WMT 2013: A tunable, accurate yet inexpensive semantic frame based mt evaluation metric," in *8th Workshop on Statistical Machine Translation (WMT 2013)*, 2013.

[36] M. Macháček and O. Bojar, "Results of the WMT13 metrics shared task," in *Eighth Workshop on Statistical Machine Translation (WMT 2013)*, Sofia, Bulgaria, August 2013.

[37] H.-P. Zhang, H.-K. Yu, D.-Y. Xiong, and Q. Liu, "Hhmm-based chinese lexical analyzer ictclas," in *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing - Volume 17*, ser. SIGHAN '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 184–187. [Online]. Available: http://dx.doi.org/10.3115/1119250.1119280

FBK's Machine Translation and Speech Translation Systems for the IWSLT 2014 Evaluation Campaign

Nicola Bertoldi[1], *Prashant Mathur*[1,2], *Nicholas Ruiz*[1,2], *Marcello Federico*[1]

[1]Fondazione Bruno Kessler
Human Language Technologies
Trento, Italy

[2]University of Trento
ICT Doctoral School
Trento, Italy

Abstract

This paper describes the systems submitted by FBK for the MT and SLT tracks of IWSLT 2014. We participated in the English-French and German-English machine translation tasks, as well as the English-French speech translation task. We report improvements in our English-French MT systems over last year's baselines, largely due to improved techniques of combining translation and language models, and using huge language models. For our German-English system, we experimented with a novel domain adaptation technique. For both language pairs we also applied a novel word trigger-based model which shows slight improvements on English-French and German-English systems. Our English-French SLT system utilizes MT-based punctuation insertion, recasing, and ASR-like synthesized MT training data.

1. Introduction

FBK's machine translation activities in the IWSLT 2014 Evaluation Campaign focused on the speech recognition and translation of TED Talks[1], a collection of public speeches on a variety of topics and with transcriptions available in multiple languages. In this paper, we describe our participation in the English-French and German-English Machine Translation tasks as well as in the English-French Spoken Language Translation task.

After a brief introduction to the baseline MT system in Section 2 employed for all tasks, in Section 3 we overview the data selection techniques used to extract TED-related data from the available huge and generic monolingual and bilingual corpora. Then, in Section 4 we describe the methods applied to combine translation models, reordering models, and language models trained on multiple corpora. Sections 5-7 give details about the actual MT and SLT systems built for evaluation task.

2. Baseline SMT system

All our task-specific systems rely on the well-known and state-of-the-art phrase-based Moses toolkit [1]; and exploit the huge amount of parallel and monolingual training data provided by the organizers. Our common baseline system features a statistical log-linear model including a phrase-based translation model (TM), a lexicalized phrase-based reordering models (RM), one or more language models (LMs), as well as distortion, word and phrase penalties.

Tuning of the baseline system is performed on tst2010 by optimizing BLEU using Minimum Error Rate Training [2]. However, all available development data sets, namely dev2010 and tst2010-2012, are included in the in-domain training data to build the systems actually employed for the 2014 evaluation campaign. The task-specific systems differ in the way training data are processed and filtered, and how the models are trained and combined.

3. Data Filtering

The idea of data selection is to find the subset of sentences within an out-of-domain corpus that better fits with a given in-domain corpus. To this purpose, we follow the procedure described in [3], implementing the bilingual cross-entropy difference [4], i.e. an adaptation of the cross-entropy difference scoring technique introduced by [5] toward bitext data selection, by means of XenC toolkit [6].

First, all sentence pairs of the out-of-domain corpus are associated with source- and target-side scores, each of which are computed as the basic technique proposes for the corresponding monolingual scenarios. We use the in-domain (TED) data as a seed and LMs of order 2.[2] Then, the sentences are sorted according to the sum of these two scores. Finally, the optimal split between useful and useless sentences is found by minimizing the source-side perplexity of a development set on growing percentages of the sorted corpus. In our experiments, dev2010 and tst2010 are concatenated and used as the filtering development set.

4. Domain Adaptation

In this section, we summarize several well-known techniques for domain adaptation we applied to build high-performance models for our SMT submissions.

[1]http://www.ted.com/talks

[2]This small LM order permits a very fast computation of the scores, without losing performance.

4.1. Translation model combination

Three methods are applied in our submissions to combine the TM built on the available parallel training corpora: namely, fill-up [7, 8], back-off, and interpolation.

4.1.1. Fill-up

In the fill-up approach, out-of-domain phrase pairs that do not appear in an in-domain (TED) phrase table are added, along with their scores – effectively filling the in-domain table with additional phrase translation options. The fill-up process is performed in a cascaded order, first filling in missing phrases from the corpora that are closest in domain to TED. Moreover, out-of-domain phrase pairs with more than four source tokens are pruned.

Following [7, 8] the fill-up approach adds k-1 provenance binary features to weight the importance of out-of-domain data, where k is the number of phrase tables to combine.

4.1.2. Back-off

The back-off approach works similarly to the fill-up technique, but does not add any provenance binary features.

4.1.3. Linear interpolation

Linear interpolation of component models is a widely used approach for building a domain adapted multi-model. Approaches such as using monolingual data or pairwise ranking optimization to set interpolation weights [9, 10], perplexity minimization [11], and combining lemmatized and non-lemmatized models [12] have been used in the past for improved domain adaptation. In this paper, we leverage a recent work of [13] which exploits the use of source-side of the parallel in-domain corpus for domain adaptation. This approach calculates a similarity score (known as BLEU-PT) for each of the out-domain translation models on the source in-domain data. We use these similarity scores and further normalize them by the number of phrases seen in each of the corresponding out-domain phrase tables. These normalized scores are then used as linear interpolation coefficients.

In this paper, we perform linear interpolation of out-of-domain models which results in one translation model. The in-domain translation model is then filled-up with the aforementioned interpolated out-domain translation model giving us a single domain adapted model.

4.2. Reordering model combination

All techniques available for combining the TMs can be applied straightforwardly to combine the RMs. The only difference regards the fill-up technique: the additional binary feature is discarded, since it is already present in the corresponding filled-up TM. Hence, a filled-up RM is exactly the same as a backed-off RM.

4.3. Language model combination

Language models are built from the monolingual training data, as well as the target language of the parallel data. As the corpora available in the IWSLT evaluation come from a number of sources, we apply several methods to combine the LMs built on the available target language training corpora, rather than concatenating the data.

4.3.1. Mixture

Monolingual subcorpora can be combined into one mixture language model [14] by means of the IRSTLM toolkit [15]. The optimization of the internal mixture weights is achieved through a cross-validation approach on the same training data; hence no external development set is required. The mixture LM type can be loaded by Moses as any other LM type.

4.3.2. Log-linear interpolation

This technique, provided directly within the Moses toolkit, consists in the log-linear interpolation of the n-gram probabilities from all component LMs. The weight optimization is performed during the tuning of all Moses features.

4.4. Factored Trigger Models

Cross-lingual lexical triggers have been already studied in natural language processing [16] and in machine translation [17]. The latter defined cross lingual triggers as a setup of a trigger word (f_i) in the source language sentence, triggering a number of words ($e_0, e_1, \ldots, e_n$) in the target language sentence. For each trigger source word f_i, we calculate pointwise mutual information (PMI) between that word and the target triggered words (e_j) as shown in Equation 1.

$$PMI_{lex}(f_i, e_j) = log \frac{P(f_i, e_j)}{P(f_i) \cdot P(e_j)} \quad (1)$$

In this paper, we extend these lexical triggers with additional factors such as POS tags and lemmas. Similar to computing the PMI for lexical triggers we compute corresponding PMIs for the POS tags and lemmas of the trigger and the triggered words in question. This is shown in the following equations:

$$PMI_{pos}(f_i, e_j) = log \frac{P(POS(f_i), POS(e_j))}{P(POS(f_i)) \cdot P(POS(e_j))} \quad (2)$$

$$PMI_{lemma}(f_i, e_j) = log \frac{P(LEM(f_i), LEM(e_j))}{P(LEM(f_i)) \cdot P(LEM(e_j))} \quad (3)$$

where $POS(x)$ is the part of speech tag of the word x and $LEM(x)$ is the lemma of the word x. These PMI are computed for all word pairs and then normalized over the whole parallel corpus. In the end, a factored trigger model (henceforth, FTM) contains three different features for each of the source/target word pair.

At decoding time, when a phrase-based machine translation system requests feature values from the FTM for a phrase pair $(f_{i,\ldots,j}, e_{k,\ldots,l})$, it returns the average sum of all the feature values for all word pairs possible within the phrase pair. Mathematically, it can be denoted as the following:

$$FTM_{lex}(f_{i,\ldots,j}, e_{k,\ldots,l}) = \sum_{z=i}^{j} \sum_{y=k}^{l} PMI_{lex}(f_z, e_y). \quad (4)$$

Similarly, POS and Lemma features are also calculated at the run-time and fed directly to the decoder providing a seamless integration of factored trigger model in a phrase based machine translation system. This integration also allows us to use any tuning algorithm (e.g. MERT, MIRA) easily.

5. English-French MT task

In order to adapt the English-French MT system to the TED-specific domain and genre, as well as to reduce the size of the models, data selection (see Section 3) is carried out on several parallel English-French corpora provided by the organizers, namely Europarl, CommonCrawl, UN, News Commentary, News Crawl, and Giga, and using the whole WIT[3] [18] training corpus as in-domain data.

Different amount of texts were selected from each corpus ranging from 2% to 30%, which are concatenated together to build one large parallel corpus containing 2.6M sentences for a total of 57M English and 63M French running words.

The system for FBK primary submission is built as follows. Two TMs and two RMs are trained independently on the parallel in-domain and selected data, using the standard Moses procedure and MGIZA++ toolkit [19] for word-alignment; TMs and RMs were combined using the back-off technique (for both TM and RM), taking WIT[3] as the primary component, for a total of 168M phrase pairs.

The French side of the in-domain and selected data are also employed to estimate a two-component mixture language model (see Section 4.3). A second huge French LM is estimated as an 8-component mixture on all permitted monolingual French data: namely, the target side of the parallel training corpora,[3] consisting of about 1.4G running words. Both LMs have order 5 and are smoothed by means of the interpolated Improved Kneser-Ney method [20]; they include 57M and 661M 5-grams, respectively. Finally, the three additional features provided by the factored trigger model (see Section 4.4) are included in the log-linear combination.

Minimum Bayes Risk (MBR) [21] decoding is applied with its default values.

As already mentioned in Section 2, all available development data sets, namely dev2010 and test2010-2012, are included in the in-domain training data to build the primary system.

[3]The monolingual French Gigaword Third Edition replaces the French side of the parallel Giga English-French corpus employed in the TM and RM model training.

In order to evaluate the contribution of the individual components of the FBK system, we submitted several contrastive runs.

- contrastive-7: derived from primary system, this system does not exploit the factored trigger model;
- contrastive-6: derived from contrastive-7, this system exploits the stack decoding instead of the MBR decoding;
- contrastive-5: derived from contrastive-6, this system does not exploit the huge French LM.

Moreover, we submitted 4 runs (contrastive 1-4) which differ from contrastive 5-7 and the primary run just in one aspect: contrastive 1-4 do not include the development data sets in the training data. The aim was to measure the impact of a limited amount of additional TED talks on the translation quality.

Finally a ninth run (contrastive-9) was submitted with a system built on top of the primary, which tests the assumption made during translation modeling that each of the features in the translation model are independent from one another. Generalized linear models can be constructed in a manner that models interactions between predictors (e.g. [22]). As a preliminary experiment, we test for interactions between the forward and backward phrase probabilities in our phrase table, expressed as a multiplication between the log probabilities.

Several observations can be drawn from the analysis of the figures reported in Table 1, also supported from preliminary experiments performed during the development phase.[4]

- The biggest performance improvement is due to the use of the large French LMs.
- MBR decoding gives a small but consistent boost in quality with respect to the stack decoding at the expense of a limited increase of decoding time.
- The factored trigger model gives a limited, sometimes negligible, improvement.
- The addition of the dev and test data has little and inconsistent impact; for tst2014 it slightly tends to improve performance, vice-versa for tst2013. This behavior is probably due to small differences among the data sets; we will investigate this issue, when we will get the references.
- Our first experiment testing for interactions suggests that the discriminative model performs better under the assumption that each phrase table feature is independent from one another.

[4]During the system development many more combinations of the considered elements were tested.

task	run	tst2013		tst2014	
		BLEU	TER	BLEU	TER
MT En-Fr	pr	38.20	44.83	34.24	46.75
	cn7	38.13	44.83	34.18	46.61
	cn6	37.88	45.05	33.79	47.02
	cn5	36.27	47.48	32.07	50.02
	cn4	38.16	44.90	33.98	47.03
	cn3	38.04	44.93	34.02	46.87
	cn2	37.95	45.08	33.67	47.24
	cn1	36.73	46.44	32.49	48.81
	cn9	37.89	44.98	34.03	46.86
MT De-En	pr	25.45	55.59	20.52	63.54
	cn	25.76	55.80	20.37	63.37

Table 1: Case-sensitive BLEU and TER results for FBK's submissions to the English-French and German-English MT tasks.

The contrastive run 5, was also applied into the joint submission by the EU-Bridge project[5] partners; details about the EU-Bridge system are available in a companion paper [23].

6. German-English MT task

Our German-English systems are built on top of the baseline system (see Section 2. Each system contains one translation model, reordering model, language model, the factored trigger model and operation sequence model; these models are then combined in a standard log-linear fashion.

The training data is composed of several publicly available corpora provided in the IWSLT MT and the WMT 2014 translation tasks. As parallel data the following corpora were taken into account: WIT^3 (version 2014-01) (TED) [18], German-English Europarl (version 7) (EP), Common Crawl (CC), MultiUN (UN), and the News Commentary (NC) corpus as distributed by the organizers of the WMT 2014. We used all the available monolingual corpora provided by the WMT 2014 translation task. The target side of the parallel corpora is also used to train our LMs.

	unselected			selected		
		De	**En**		**De**	**En**
Corpus	**Segm**	**Words**	**Words**	**Segm**	**Words**	**Words**
TED	171K	3.3M	3.46M	171K	3.3M	3.46M
CC	2.4M	56M	58M	462K	10.5M	10.7M
EP	1.9M	52M	53M	188K	3.58M	3.64M
UN	162K	5.8M	5.66M	45K	1.59M	1.52M
NC	200K	5.25M	5.0M	59K	1.4M	1.3M

Table 2: Statistics of the parallel and monolingual data exploited for training our German-English systems. For the parallel data, statistics before and after data selection are reported. Symbols "M" and "K" stand for 10^6 and 10^3, respectively.

Table 2 shows the statistics of the German-English data. The average number of words per sentence in all of the above corpora is relatively lower on German side than on the English side. This is largely due to compounding, where Noun-Verb, Noun-Noun, Adjective-Noun pairs, for example, are combined together to form a larger compound. Models trained using raw German text could lead to a high out-of-vocabulary rate on unseen texts [24]. We leverage a trainable compound splitter [25], which splits a compound based on a frequency based metric. We train one compound splitter model on TED monolingual corpus (German) which contains 3.35M running words and another on the source (German) side of the TED parallel corpus, which contains 3.2M running words. The first splitter is `aggressive` while the second model is more `passive`. Each of the selected corpora goes through these splitter models resulting in two different systems for German-English task.

Primary: We select different amount of texts from each corpus ranging from 10% to 30% of each corpus' original size. Aggressive splitting is done on source side (German) of all training, development and test corpora. As the German-English language pair shows a high amount of reordering we have used the hierarchical phrase reordering model as described by [26]. Each system has one TM and one RM that are built on each domain, comprising a total of 5 TMs and RMs. Linear interpolation as described in Section 4.1.3 is used to combine the out-of-domain models (CC, EP, UN and NC), resulting in a single background TM and RM. The TED TM and RM are then filled-up with the background TM and RM and a binary provenance feature is added to the TM. Another model that we use is a lexically driven 5-gram operation sequence model (OSM) [27] with a standard feature set. The OSM model is built on the concatenation of all five parallel corpora. As the factored trigger model usually results in a big phrase table, we use just the TED domain to build the model. TreeTagger [28] assigns a lemma and POS-tag to each word which are included as two factors in the factored trigger model.

Contrastive: The contrastive system is configured similarly to the Primary system, except that we use the passive splitter model to split the German compounds.

Evaluation results show that both systems are at par with one other on 2013 and 2014 test sets. On comparing just the BLEU scores on both test sets, we see that a passive splitter is useful for 2013 test set while an aggressive splitting is required on the 2014 test set. The factored trigger model was useful for German-English pair; an offline evaluation on the development set (tst2010) showed that the primary system with the FTM gave a jump of 0.2 BLEU points over the system where we do not use FTM.

7. English-French SLT task

The sections below describe the steps followed to perform English-French speech translation. Each of the submitted translations are drawn from machine translation systems derived from the contrastive-6 MT system (Section 5), which uses stack decoding. We briefly describe the techniques applied to normalize and preprocess the ASR outputs to make

[5] `http://www.eu-bridge.eu`

them suitable for translation. We additionally provide a brief summary of a text normalization technique relying on phonemic confusion to synthesize ASR outputs for MT training. Finally, we describe our experimental results.

7.1. Preprocessing

Prior to translating ASR outputs, we perform the following normalization steps to make them compatible with our phrase-based SMT system.

Similar to the MT track, we tokenize ASR outputs using the scripts provided by Moses. After tokenization, we recase the outputs. The recaser system is trained using the Moses scripts and a 3-gram LM. The recaser model and language models are trained on a concatenation of TED and WMT News Commentary data. Finally, we insert punctuation via monotonic machine translation, similar to the approach of [29].

7.2. Phoneme-motivated Text Normalization

A SMT system trained only on transcripts and other text data results yields a search space that is inaccessible by ASR outputs that may contain errors and text normalization issues. In an ideal scenario, we would train our spoken language translation system on a combination of text corpora and speech recognition outputs with reference translations; however, a sufficiently large amount of such speech corpora is not readily available. In order to make our machine translation system more tolerant of potential ASR errors, we use a similar phoneme-motivated text normalization approach as outlined in our previous year's submission [30] to generate additional bilingual training data from the text corpora provided in the evaluation.

We adapt the MT training data into ASR-like output to anticipate ASR errors and text normalization issues during SMT model training. We do this by leveraging several components from a target ASR system. In our experiments, we use the FBK's Kaldi English ASR system, which was used in our ASR submission [31]. Similar to [32], we transform the text corpora into synthetic ASR outputs by first converting the text corpora into phonemes and then "translating" each phoneme sequence back into words that more closely match the output of our ASR system. Following the exposition described in [30], we use the Festival text-to-speech engine[6] to convert each word in our ASR system's pronunciation lexicon into phoneme sequences. The word to phoneme sequence mappings are used to generate a phrase table that translates from phoneme sequences to words. We augment the word to phoneme sequence mappings with the original pronunciation entries in the ASR lexicon. We assign uniform forward and backward phrase probabilities to each phoneme sequence to word mapping in the phrase table and omit the lexical probabilities from the model. We use the phrase table and the original ASR system's 4-gram English language model [31] as components in a Moses phrase-based SMT system.

[6] http://www.cstr.ed.ac.uk/projects/festival

run	BLEU	TER
pr	25.39	59.53
cn1	25.29	59.64
cn2	25.08	60.15
cn3	24.23	61.63
cn4	24.28	61.65
cn5	24.00	62.02

Table 3: Case-sensitive BLEU and TER results for FBK's tst2014 submissions to the English-French SLT task.

The system is tuned on the tst2010 data set: the reference transcript is converted to phonemes using the TTS system described above. Since our goal is to convert clean transcripts into synthetic ASR output, it serves as our source text. Our reference set consists of the 1-best ASR outputs from our best Kaldi ASR system, which transcribed the audio corresponding to the tst2010 transcripts. Tuning is performed to optimize BLEU via MERT.

After tuning, we convert all of the out-of-domain text corpora, aside from Common Crawl, into ASR-like output using the trained system. Each ASR-like corpus is tokenized and recased according to the steps described above. The new damaged corpora are concatenated together and used to train an English-French phrase table and reordering model, using the same training pipeline as described in Section 2. After the phrase table and reordering models are trained, we use the fill-up technique with the models trained in the MT task (Section 5).

We additionally train a monotonic phoneme-to-phoneme phrase-based SMT system to generate additional confusable pronunciations for each of the lexical entries, using a 4-gram phoneme language model and the default Moses parameters. The training is performed in a similar manner as in [32].

7.3. Experiments

We submitted six alternative translations of the ASR outputs on tst2014. Our first set of translations (pr, cn1, cn2) use the 1-best ROVER system combination provided by the organizers. Our primary system uses all of the techniques listed above. Our first contrastive system (cn1) omits the phoneme-to-phoneme pronunciation generation. Our second contrastive system (cn2) does not include any synthetic phrase table entries. Our second set of translations (cn3-5) use the same sequence of steps as those listed above. Rather than using the ROVER ASR hypothesis, we use the ASR hypothesis corresponding to FBK's primary submission in the English ASR track. Results are shown in Table 3.

In particular, we note an increase of 1 BLEU by using the ROVER outputs instead of FBK's primary system. Additionally, we see an improvement of approximately 0.3 BLEU when using our phoneme-based text normalization

techniques.

8. References

[1] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic, 2007, pp. 177–180. [Online]. Available: http://aclweb.org/anthology-new/P/P07/P07-2045.pdf

[2] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, E. Hinrichs and D. Roth, Eds., 2003, pp. 160–167. [Online]. Available: http://www.aclweb.org/anthology/P03-1021.pdf

[3] M. Cettolo, C. Servan, N. Bertoldi, M. Federico, L. Barrault, and H. Schwenk, "Issues in Incremental Adaptation of Statistical MT from Human Post-edits," in *Proceedings of the MT Summit XIV Workshop on Post-editing Technology and Practice (WPTP-2)*, Nice, France, September 2013, pp. 111–118.

[4] A. Axelrod, X. He, and J. Gao, "Domain Adaptation via Pseudo In-Domain Data Selection," in *Conference on Empirical Methods in Natural Language Processing*, Edinburgh, United Kingdom, 2011, pp. 355–362.

[5] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *ACL (Short Papers)*, 2010, pp. 220–224.

[6] A. Rousseau, "Xenc: An open-source tool for data selection in natural language processing," *The Prague Bulletin of Mathematical Linguistics*, vol. 100, pp. 73–82, 2013.

[7] P. Nakov, "Improving English-Spanish Statistical Machine Translation: Experiments in Domain Adaptation, Sentence Paraphrasing, Tokenization, and Recasing," in *Workshop on Statistical Machine Translation, Association for Computational Linguistics*, 2008.

[8] A. Bisazza, N. Ruiz, and M. Federico, "Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation," in *International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011, pp. 136–143.

[9] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," in *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 128–135. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0217

[10] B. Haddow, "Applying pairwise ranked optimisation to improve the interpolation of translation models," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, Georgia, USA, June 2013, pp. 342–347.

[11] R. Sennrich, "Perplexity minimization for translation model domain adaptation in statistical machine translation," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Avignon, France: Association For Computational Linguistics, 2012, pp. 539–549. [Online]. Available: http://dx.doi.org/10.5167/uzh-61712

[12] R. Zhang and E. Sumita, "Boosting statistical machine translation by lemmatization and linear interpolation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 181–184. [Online]. Available: http://www.aclweb.org/anthology/P07-2046

[13] P. Mathur, S. Venkatapathy, and N. Cancedda, "Fast domain adaptation of smt models without in-domain parallel data," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, August 2014, pp. 1114–1123. [Online]. Available: http://www.aclweb.org/anthology/C14-1105

[14] M. Federico and R. De Mori, "Language modelling," in *Spoken Dialogues with Computers*, R. D. Mori, Ed. London, UK: Academy Press, 1998, ch. 7, pp. 199–230.

[15] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models," in *Proceedings of Interspeech*, Brisbane, Australia, 2008, pp. 1618–1621.

[16] W. Kim and S. Khudanpur, "Lexical triggers and latent semantic analysis for cross-lingual language model adaptation," *ACM Transactions on Asian Language Information Processing*, vol. 3, no. 2, pp. 94–112, June 2004. [Online]. Available: http://doi.acm.org/10.1145/1034780.1034782

[17] C. Lavecchia, K. Smaïli, D. Langlois, and J.-P. Haton, "Using inter-lingual triggers for Machine translation," in *8th Annual Conference of the International Speech Communication Association - INTERSPEECH 2007*.

Antwerp, Belgium: ISCA, Aug. 2007, pp. 2829–2832. [Online]. Available: http://hal.inria.fr/inria-00155791

[18] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of the Annual Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012. [Online]. Available: http://hltshare.fbk.eu/EAMT2012/html/Papers/59.pdf

[19] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, ser. SETQA-NLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 49–57. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622110.1622119

[20] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Harvard University, Tech. Rep. TR-10-98, 1998.

[21] S. Kumar and W. Byrne, "Minimum bayes-risk decoding for statistical machine translation," in *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2004.

[22] M. L. Buis, "Stata tip 87: Interpretation of interactions in nonlinear models," *Stata Journal*, vol. 10, no. 2, pp. 305–308(4), 2010. [Online]. Available: http://www.stata-journal.com/article.html?article=st0194

[23] M. Freitag, J. Wuebker, S. Peitz, M. Huck, A. Birch, N. Durrani, P. Koehn, M. Mediani, I. Slawik, J. Niehues, E. C. A. Waibel, N. Bertoldi, M. Cettolo, and M. Federico, "Combined spoken language translation," in *Proc. of the International Workshop on Spoken Language Translation*, Lake Tahoe, California, USA, December 2014, p. to appear.

[24] N. Ruiz, A. Bisazza, R. Cattoni, and M. Federico, "FBK's Machine Translation Systems for IWSLT 2012's TED Lectures," in *International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012, pp. 61–68.

[25] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL)*, 2003.

[26] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Morristown, NJ, USA: Association for Computational Linguistics, 2008, pp. 848–856.

[27] N. Durrani, A. Fraser, H. Schmid, H. Hoang, and P. Koehn, "Can markov models over minimal translation units help phrase-based smt?" in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 399–405. [Online]. Available: http://www.aclweb.org/anthology/P13-2071

[28] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proceedings of International Conference on New Methods in Language Processing*, 1994.

[29] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling punctuation prediction as machine translation," in *International Workshop on Spoken Language Translation*, San Francisco, CA, USA, Dec. 2011, pp. 238–245. [Online]. Available: http://www.mt-archive.info/10/IWSLT-2011-Peitz.pdf

[30] A. Aue, Q. Gao, H. Hassan, X. He, G. Li, N. Ruiz, and F. Seide, "Msr-fbk iwslt 2013 slt system description," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, December 2013.

[31] B. Babaali, R. Serizel, S. J. D. Falavigna, R. Gretter, and D. Giuliani, "FBK @ IWSLT 2014 - ASR track," in *Proc. of the International Workshop on Spoken Language Translation*, Lake Tahoe, California, USA, December 2014, p. to appear.

[32] Q. F. Tan, K. Audhkhasi, P. G. Georgiou, E. Ettelaie, and S. S. Narayanan, "Automatic speech recognition system channel modeling." in *INTERSPEECH*, 2010, pp. 2442–2445.

Edinburgh SLT and MT System Description for the IWSLT 2014 Evaluation

Alexandra Birch, Matthias Huck, Nadir Durrani, Nikolay Bogoychev, Philipp Koehn

School of Informatics
University of Edinburgh
Scotland, United Kingdom
a.birch@ed.ac.uk {mhuck,dnadir,nbogoych,pkoehn}@inf.ed.ac.uk

Abstract

This paper describes the University of Edinburgh's spoken language translation (SLT) and machine translation (MT) systems for the IWSLT 2014 evaluation campaign. In the SLT track, we participated in the German↔English and English→French tasks. In the MT track, we participated in the German↔English, English→French, Arabic↔English, Farsi→English, Hebrew→English, Spanish↔English, and Portuguese-Brazil↔English tasks.

For our SLT submissions, we experimented with comparing operation sequence models with bilingual neural network language models. For our MT submissions, we explored using unsupervised transliteration for languages which have a different script than English, in particular for Arabic, Farsi, and Hebrew. We also investigated syntax-based translation and system combination.

1. Introduction

The University of Edinburgh's translation engines are based on the open source Moses toolkit [1]. We set up phrase-based systems [2] for all SLT and MT tasks covered in this paper, and additionally a string-to-tree syntax-based system [3, 4] for the English→German MT task.

The setups for our phrase-based systems have evolved from the configurations of the engines we built for last year's IWSLT [5] and for this year's Workshop on Statistical Machine Translation (WMT) [6]. The notable features of these systems are:

- Phrase translation scores in both directions, smoothed with Good-Turing discounting
- Lexical translation scores in both directions
- Word and phrase penalties
- Six simple count-based binary features
- Phrase length features
- Distance-based distortion cost
- A hierarchical lexicalized reordering model [7]
- Sparse lexical and domain indicator features [8]
- Operation sequence models (OSMs) over different word representations [9, 10]
- A 5-gram language model (LM) over words

We typically train factored phrase-based translation models [11, 12] and also incorporate higher order *n*-gram LMs over word representations given by the factors. Factors can for instance be lemma, part-of-speech (POS) tag, morphological tag, or automatically learnt word classes in the manner of Brown clusters [13].

Edinburgh's syntax-based systems have recently yielded state-of-the-art performance on English→German news translation tasks [14, 15] but have not been applied in an IWSLT-style setting before. Standard features of our string-to-tree syntax-based systems are:

- Rule translation scores in both directions, smoothed with Good-Turing discounting
- Lexical translation scores in both directions
- Word and rule penalties
- A rule rareness penalty
- The monolingual PCFG probability of the tree fragment from which the rule was extracted
- A 5-gram LM over words

For our Spanish↔English and Portuguese-Brazil↔English submissions, we ran the engines as described in last year's system description paper [5]. In the following, we focus on describing the new systems which were developed for the rest of the tasks.

Our this year's IWSLT systems were trained using monolingual and parallel data from WIT^3 [16], Europarl [17], MultiUN [18], the Gigaword corpora as provided by the Linguistic Data Consortium [19], the German Political Speeches Corpus [20], and the corpora provided for the WMT shared translation task [21].

Word alignments for the MT track systems were created by aligning the data in both directions with MGIZA++ [22] and symmetrizing the two alignments with the grow-diag-final-and heuristic [23, 2]. Word alignments for the SLT track systems were created using fast_align [24].

The SRILM toolkit [25] was employed to train 5-gram language models (LMs) with modified Kneser-Ney smoothing [26]. We trained individual LMs on each corpus and then interpolated them using weights tuned to minimize perplexity on a development set. KenLM [27] was utilized for LM scoring during decoding. Model weights for the log-linear

model combination [28] were optimized with batch k-best MIRA [29] to maximize BLEU [30]. Where not otherwise stated, the systems were tuned on dev2010.

Besides participating in the evaluation campaign with our individual engines, we also collaborated with partners from the EU-BRIDGE project to produce additional joint submissions. The combined systems of the University of Edinburgh, RWTH Aachen University, Karlsruhe Institute of Technology, and Fondazione Bruno Kessler are described in [31].

2. Spoken Language Translation

Edinburgh's spoken language translation system experiments set out to compare two recent strands of research in terms of their performance and their properties in order to understand the contributions of each. The first strand of research is bilingual neural network langauge models. There has recently been a great deal of interest bilingual neural network language models as they have shown strong gains in performance for Arabic→English, and to a lesser extent for Chinese→English [32]. It is still not clear what the exact contribution of the bilinugal language model is, and there is reason to believe that its contribution may be that it allows the SMT model to overcome strong phrase pair independence assumptions.

The second strand of research is operation sequence modelling [33, 34]. The integration of the OSM model into phrase-based decoding directly addresses the problem of the phrasal independence assumption by modelling the context of phrase pair translations. We aim to compare these two different approaches and combining them. As we see, combining OSM and the bilingual NN language model slightly outperforms all other models, including the state-of-the-art OSM model, but only for English→French and only very slightly.

2.1. Baseline

For the SLT track, we trained phrase-based models using Moses with mostly default settings. We further included basic sparse features [35] and we used factors. For German→English we used POS tags, morphological tags and lemmas as factors in decoding [11], and for English→German we used POS tags and morphological tags on the target side. Table 1 lists the factors used for the translation model, and the factors over which we trained OSM models.

The SLT and the MT systems were trained in a similar fashion, with the main difference being that for SLT no pre-reordering was performed for German→English as this relies on grammatically correct test sentences, and automatic speech recognition (ASR) output, especially for German, is difficult to parse correctly. We trained the SLT systems on the Europarl, WIT^3, News Commentary, and Commoncrawl corpora. The monolingual data contained the target side of the parallel corpora, the news language model data provided for WMT, and the LDC Gigaword for French and English. The number of words of training data can be seen in Table 1.

	EN→FR	EN→DE	DE→EN
Src Factors	w,c	w,c	w,l,p,m
Tgt Factors	w,c	w,p,m,c	w,l,p
OSM	w,c	w,c	w,l,p and m→p
No. words ‖	138M/153M	116M/110M	110M/116M
No. words mono	2673M	2214M	6600M

Table 1: SLT track: Factors used by translation models and OSM models (words w, clusters c, lemma l, pos p, morphology m) and the size of the parallel and monolingual training data in millions of words.

2.2. Monolingual Punctuation Models

One of the main challenges of spoken language translation is to overcome the mismatch in the style of data that the speech recognition systems output, and the written text that is used to train the translation model. ASR system output lacks punctuation and capitalisation and this is one of the main stylistic differences. Previous research [36, 5] suggests that it is preferrable to punctuate the text before translation, which is what we did by training a monolingual translation system for our two source languages: German and English. The "source language" of the punctuation model has punctuation and capitalisation stripped, and the "target language" is the full original written text. Our handling of punctuation uses a phrase-based translation model with no distortion or reordering, and we tuned the model to the ASR input text (dev2010 for English, and dev2012 for German) using batch MIRA and the BLEU score. After running ASR output through the punctuation model, it is then translated with a standard machine translation model, trained directly on the parallel written text, in a very similar fashion to the MT system, except that for our official submission we tuned the MT model to the ASR tuning set.

2.3. Operation Sequence Model

We investigated applying a number of OSM models [33, 34] to the basic phrase-based translation model. OSM addresses the problem of the phrasal independence assumption since the model considers context beyond phrasal boundaries. The OSM model represents a bilingual sentence pair and its alignment through a sequence of operations that generate the aligned sentence pair. An operation either generates source and target words or it performs reordering by inserting gaps and jumping forward and backward. It has shown to improve performance over many language pairs, and to help even more when sequence models are applied over more general factors such as POS tags and GIZA++'s `mkcls` clusters [5]. For this experiment we applied the best OSM settings from last year's IWSLT experiments which included models over words, lemmas, POS tags, and clusters depending on the language pair. See Table 1 for details.

2.4. Bilingual Neural Network Language Model

There has recently been a great deal of interest in including neural networks in machine translation [37, 38]. There is hope that neural networks provide a way to relax some of the more egregious independence assuptions made in translation models. The challenge with neural networks however, is that they are computationally very expensive, and getting them to operate at scale requires sophisticated efficiency techniques. A recent paper which was able to fully integrate a neural network which includes both source side and target side context in decoding [32], and they managed to show big improvements for a small Arabic→English task, and smaller improvements for a Chinese→English task. We implemented a bilingual neural network language model in order to investigate what their benefits are to state-of-the-art translation models.

We implemented a BiNNLM as a feature function inside Moses, following closely the implementation outlined in [32]. The main focus of our design is to make the Moses specific code flexible and independent of the neural network language model that would be used for scoring. As a result any NNLM could implement the interface and be used by Moses during decoding. Some features such as backoff to POS tag in case of unknown word or use of special $< null >$ token to pad an incomplete parse in the chart decoder are made optional. Currently the implemented backends are NPLM [39] and OxLM [40]. Implementation is available for both phrase based and hierarchical Moses. For our experiments we chose NPLM to be our NNLM backend. We chose it, because it features noise contrastive estimation (NCE) which allows us to avoid having to apply softmax to normalize the outputs, as it is infeasible to do so with large vocabularies. Another benefit of NPLM is that when using NCE and a neural network with one hidden layer we can precompute the values for the first hidden layer of all vocabulary terms, similarly to what [32] do. We also modified the NPLM code a bit and used Magma enabled fork of the Eigen library[1] to speed up the training. This results in a decoder which is about twice as slow as the phrase-based decoder without BiNNLM On average decoding speed is three sentences per second when using BiNNLM, which highlights that this implementation is fast enough to make large experiments possible.

For these experiments we used a target context of four words, and an aligned source window of nine words. Note that NPLM does not support separate source and target contexts so what we did is use the parallel corpora to extract 14-grams which consist of 9 source and 5 target words. Once those 14-grams are extracted we train NPLM on them as if it were a monolingual dataset. The size of our word embedding layers was 256 for the EN→FR, and 150 for DE→EN language models. Increasing the size of the embeddings for DE→EN did not increase performance, but decreasing it for EN→FR seemed to hurt performance. We used just one hidden layer to allow precomputation and much faster decoding. We used a source and target vocabulary size of 16k words, and used a part-of-speech backoff for the less frequent words for the DE→EN system, and backoff to the UNK token for EN→FR.

[1] https://github.com/bravegag/eigenmagma

	EN→FR	DE→EN
Baseline	35.7	32.5
OSM full	37.3	33.0
BiNNLM	36.7	32.4
OSM + BiNNLM	37.4	32.8

Table 2: Performance comparison of OSM and BiNNLM (average case-sensitive BLEU score of IWSLT test sets 2010-2012).

	EN→FR	DE→EN	EN→DE
dev2012	-	21.00	-
dev2010	23.39	-	21.25
test2014	25.50	17.67	17.00

Table 3: Results of submission systems in the SLT track (case-sensitive BLEU scores).

2.5. Results

Looking at Table 2 it seems that both the OSM model and the BiNNLM model outperform the baseline. The OSM model is stronger than the BiNNLM when both features are used separately. However, for the EN→FR task, combining OSM and BiNNLM outperforms OSM on its own by 0.14 BLEU points. The baseline translation systems use large amounts of parallel and monolingual data, and it is not surprising that our first attempt at using BiNNLM did not resoundingly beat the previously state-of-the-art OSM models. It is surprising perhaps that BiNNLM did much better for EN→FR than DE→EN. This is similar to the Devlin et al. result where their AR→EN improvements were much stronger than their ZH→EN results.

From the results here it does seem like the advantages gained by applying OSM and BiNNLM might overlap, given that there is not a large improvement seen when combining the two types of features.

We used the baseline systems trained with OSM models for our official submission to the IWSLT 2014 evaluation. We tuned these on the supplied ASR development sets. The results are shown in Table 3.

3. Machine Translation

This section contains a description of the experiments we carried out for tasks in the MT track of the evaluation campaign.

Pair	Training	tst2010	tst2011	tst2012
	-	26.7	26.3	29.8
AR→EN	7.6K	26.8	26.5	29.9
OOV		393	345	442
	-	8.8	9.6	9.5
EN→AR	9.1K	8.8	9.7	9.6
OOV		351	277	424
	-	15.6	20.7	15.6
FA→EN	5.5K	15.8	21.0	15.8
OOV		337	451	628
	-	30.1	31.5	31.7
HE→EN	14K	30.3	31.8	31.9
OOV		837	753	892

Table 4: Effect of unsupervised transliteration models. Training = extracted transliteration corpus (types). First rows: system without transliteration. Second rows: transliterating OOVs. Third rows: number of OOVs (types) in each test.

EN→AR	tst2010	tst2011	tst2012
baseline	8.3	8.3	8.7
+ Gigaword + UN	8.9	9.2	9.6

Table 5: Effect of Gigaword and UN monolingual data on English→Arabic translation quality.

3.1. Unsupervised Transliteration Model

Arabic, Farsi and Hebrew are written in different writing scripts as English, therefore the conventional method of copying unknown words to the output is not a good idea. We built unsupervised transliteration models [41] to translate OOV words.

The transliteration model is induced using an EM-based method [42]. We extracted transliteration pairs automatically from the word-aligned parallel data and used it to learn a transliteration system. We then built transliteration phrase-tables for translating OOV words and used the post-decoding method (Method 2 as described in the paper) to translate these. Table 4 show results from using unsupervised transliteration models. Small improvements were shown in all cases. Note that not all the OOVs can be translated correctly through transliteration. Only a handful of these were named entities and foreign words that could be transliterated.

3.2. Arabic-English MT

We carried out a number of experiments for the Arabic-English language pair which we now discuss briefly.

Tokenization. We used MADA tokenizer for source-side Arabic [43] and tried different segmentation schemes including D*, S2 and ATB. The ATB segmentation consistently outperformed other schemes.

Modified Moore and Lewis Filtering. The in-domain datasets (TED talk corpus) are small and a large out-of-domain corpus (UN) is available. We tried to explore various ways to make best use of the out-of-domain data to improve the baseline system. We used Modified Moore and Lewis as known as MML [44] filtering, to subsample training data that is similar to the in-domain data. We varied the percentage of bilingual UN data selected between 2%, 5%, 20% and 100%. Adding any percentage of UN data did not give any gains in the performance. Using 2% gave best results, however, they were still below the baseline system.

Backoff Phrase Tables. Instead of using UN data directly we used it with the *backoff* phrase-table method. This allows Moses to use the phrase-table built with the UN data only when a phrase is unknown to phrase-table trained from the in-domain data. The backoff order determines the maximum phrase length for which this operation is allowed. We used backoff order of 5. Using backoff phrase tables gave slight improvement in English→Arabic, results stayed constant or dropped in Arabic→English direction.

Class-based Model. We explored the use of automatic word clusters in phrase-based models [10]. We computed the clusters with GIZA++'s `mkcls` [45] on the source and target side of the parallel training corpus. Clusters are word classes that are optimized to reduce n-gram perplexity. By generating a cluster identifier for each output word, we are able to add an n-gram model over these identifiers as an additional scoring function. The inclusion of such an additional factor is trivial given the factored model implementation [11] of Moses. The n-gram model is trained in the similar way as the regular language model. The lexically driven OSM model falls back to very small context sizes of two to three operations due to data sparsity. Learning operation sequences over cluster-ids enables us to learn richer translation and reordering patterns that can generalize better in sparse data conditions.

Using class-based models, however, did not give any improvements for Arabic-English tasks. We also trained OSM models over cluster-ids. This result contradicts our findings in last year IWSLT paper [5] where we reported significant gains using class-based models on many European language pairs with English as source language.

Monolingual Arabic Data. Unlike parallel data, adding Gigaword and UN monolingual data in English→Arabic translation task gave significant improvements. The gains are shown in Table 5.

3.3. German→English MT

For the German→English MT task system, pre-reordering [46] and compound splitting [47] were applied to the German source language side in a preprocessing step. A factored translation model was employed. Source side factors are word, lemma, POS tag, and morphological tag. Target side factors are word, lemma, and POS tag. Supplementary to the features listed in Section 6, we

incorporated two additional LMs into the German→English MT system: a 7-gram LM over POS tags and a 7-gram LM over lemmas (both trained on WIT^3 only). Model weights were optimized on a concatenation of dev2010 and dev2012. Table 6 contains the results on the three test sets.

3.4. English→French MT

We submitted outputs of three phrase-based systems for the English→French MT task: a *primary* system and two contrastive systems (*contrastive 1* and *contrastive 2*). All available training corpora were utilized, with the exception of the MultiUN corpus and the WMT 10^9 French-English corpus, which we excluded from both the parallel and the LM training data. Our systems comprise Brown clusters with 200 classes as additional factors on source and target side. Supplementary to the features listed in Section 6, we incorporated a 7-gram LM over Brown clusters. Furthermore, a bilingual neural network language model as described in Section 2.4 was integrated into the *primary* and the *contrastive 1* system. The primary system was tuned on tst2012, the contrastive systems were tuned on dev2010.

The characteristics of the setup denoted as *contrastive 1* are thus the same as those of the primary submission. We employed identical configuration parameters and features, the only difference between the two systems is the usage of a different tuning set for the optimization of model weights. The setup denoted as *contrastive 2* is similar to *contrastive 1* but does not comprise the bilingual neural network language model. Experimental results are presented in Table 7.

3.5. English→German MT

For the English→German MT task, we submitted outputs of a phrase-based system (*primary*), a syntax-based system (*contrastive 1*), and a system combination (*contrastive 2*). Table 8 shows their respective performance in terms of BLEU scores.

Phrase-based System. The *primary* system is phrase-based with factored models. Source side factors are word, POS tag, and Brown cluster (2000 classes). Target side factors are word, POS tag, Brown cluster (2000 classes), and morphological tag. The primary system was trained with all corpora. Additional features of the primary system are: a 5-gram LM over Brown clusters, a 7-gram LM over morphological tags, and a 7-gram LM over POS tags. Model weights of the primary system were optimized on a concatenation of dev2010 and dev2012.

We trained a second, smaller phrase-based system on in-domain bitexts only (i.e., we restricted the parallel training data to the WIT^3 corpus). We denote this second phrase-based system as *phrase-based in-domain*. Individual hypotheses from the phrase-based in-domain system have not been submitted for the evaluation; we merely added them as auxiliary inputs to our system combination. Additional features of the phrase-based in-domain system are: a 5-gram LM over Brown clusters and a 7-gram LM over morphological tags (the latter trained on WIT^3 only). Model weights of the phrase-based in-domain system were optimized on dev2010.

DE→EN	**tst2010**	**tst2011**	**tst2012**
primary	31.6	37.3	31.7

Table 6: Results for the German→English MT task (case-sensitive BLEU scores).

EN→FR	**tst2010**	**tst2011**	**tst2012**
primary	34.4	41.5	44.9
contrastive 1	33.8	40.3	41.4
contrastive 2	33.6	40.2	41.0

Table 7: Results for the English→French MT task (case-sensitive BLEU scores). The contrastive systems were tuned on dev2010, the primary system was tuned on tst2012. A bilingual neural network language model was integrated into *primary* and *contrastive 1*.

EN→DE	**tst2010**	**tst2011**	**tst2012**
phrase-based (primary)	24.9	27.8	23.4
phrase-based in-domain	24.1	26.7	22.2
syntax-based (contrastive 1)	24.8	26.5	23.1
syscom (contrastive 2)	26.0	27.8	24.5

Table 8: Results for the English→German MT task (case-sensitive BLEU scores). The *contrastive 2* submission is a system combination of three systems which was tuned on tst2012.

Syntax-based System. The *contrastive 1* system is a string-to-tree translation system with similar features as the ones described in [15]. The target-side data was parsed with BitPar [48], and right binarization was applied to the parse trees. The system was adapted to the TED domain by extracting separate rule tables (from the WIT^3 corpus and from the rest of the parallel data) and merging them with a fill-up technique [49]. Augmenting the system with non-syntactic phrases [50] and adding soft source syntactic constraints [51] yielded further improvements. Model weights of the syntax-based system were optimized on a concatenation of dev2010 and dev2012.

System Combination. We combined the outputs of the phrase-based primary system, the auxiliary phrase-based in-domain system, and the string-to-tree syntax-based system with the MT system combination approach implemented in the Jane toolkit [52]. The parameters of the system combination were optimized on tst2012. The consensus translation produced by the system combination (syscom) was submitted as *contrastive 2*.

4. Summary

The Edinburgh submissions for IWSLT cover many language pairs and research techniques. We have implemented a bilingual neural network language model feature in Moses and have demonstrated that it can lead to state-of-the-art results for English→French. BiNNLM seems less beneficial for German→English, however. Our experiments further confirmed the benefit of using OSM, transliteration and system combination.

5. Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658 (EU-BRIDGE) and grant agreement n° 288487 (MosesCore).

6. References

[1] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. J. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic, June 2007, pp. 177–180.

[2] P. Koehn, F. J. Och, and D. Marcu, "Statistical Phrase-Based Translation," in *Proc. of the Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics (HLT-NAACL)*, Edmonton, Canada, May/June 2003, pp. 127–133.

[3] M. Galley, M. Hopkins, K. Knight, and D. Marcu, "What's in a translation rule?" in *Proc. of the Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics (HLT-NAACL)*, Boston, MA, USA, May 2004, pp. 273–280.

[4] P. Williams and P. Koehn, "GHKM Rule Extraction and Scope-3 Parsing in Moses," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*, Montreal, Canada, June 2012, pp. 434–440.

[5] A. Birch, N. Durrani, and P. Koehn, "Edinburgh slt and mt system description for the IWSLT 2013 evaluation," in *Proceedings of the 10th International Workshop on Spoken Language Translation*, Heidelberg, Germany, December 2013, pp. 40–48.

[6] N. Durrani, B. Haddow, P. Koehn, and K. Heafield, "Edinburgh's phrase-based machine translation systems for WMT-14," in *Proceedings of the ACL 2014 Ninth Workshop on Statistical Machine Translation*, Baltimore, MD, USA, June 2014, pp. 97–104.

[7] M. Galley and C. D. Manning, "A Simple and Effective Hierarchical Phrase Reordering Model," in *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, Honolulu, HI, USA, Oct. 2008, pp. 847–855.

[8] E. Hasler, B. Haddow, and P. Koehn, "Sparse Lexicalised Features and Topic Adaptation for SMT," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, Dec. 2012, pp. 268–275.

[9] N. Durrani, H. Schmid, and A. Fraser, "A joint sequence translation model with integrated reordering," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, June 2011, pp. 1045–1054.

[10] N. Durrani, P. Koehn, H. Schmid, and A. Fraser, "Investigating the Usefulness of Generalized Word Representations in SMT," in *Proceedings of the 25th Annual Conference on Computational Linguistics (COLING)*, Dublin, Ireland, August 2014, pp. 421–432.

[11] P. Koehn and H. Hoang, "Factored translation models," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 868–876.

[12] P. Koehn and B. Haddow, "Interpolated Backoff for Factored Translation Models," in *Proc. of the Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, San Diego, CA, USA, Oct./Nov. 2012.

[13] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes," in *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 1999, pp. 71–76.

[14] M. Nadejde, P. Williams, and P. Koehn, "Edinburgh's Syntax-Based Machine Translation Systems," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Sofia, Bulgaria, Aug. 2013, pp. 170–176.

[15] P. Williams, R. Sennrich, M. Nadejde, M. Huck, E. Hasler, and P. Koehn, "Edinburgh's Syntax-Based Systems at WMT 2014," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Baltimore, MD, USA, June 2014, pp. 207–214.

[16] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[17] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proc. of the MT Summit X*, Phuket, Thailand, Sept. 2005.

[18] A. Eisele and Y. Chen, "MultiUN: A Multilingual Corpus from United Nation Documents," in *Proceedings of the Seventh conference on International Language Resources and Evaluation*, May 2010, pp. 2868–2872.

[19] Linguistic Data Consortium (LDC), http://www.ldc.upenn.edu.

[20] A. Barbaresi, "German Political Speeches, Corpus and Visualization," ENS Lyon, Tech. Rep., 2012, 2nd Version. [Online]. Available: http://purl.org/corpus/german-speeches

[21] Shared Translation Task of the ACL 2014 Ninth Workshop on Statistical Machine Translation, http://www.statmt.org/wmt14/translation-task.html.

[22] Q. Gao and S. Vogel, "Parallel Implementations of Word Alignment Tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, ser. SETQA-NLP '08, Columbus, OH, USA, June 2008, pp. 49–57.

[23] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[24] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," in *Proc. of the Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics (HLT-NAACL)*, Atlanta, GA, USA, June 2013, pp. 644–648.

[25] A. Stolcke, "SRILM – an Extensible Language Modeling Toolkit," in *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP)*, Denver, CO, USA, Sept. 2002, pp. 901–904.

[26] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," Computer Science Group, Harvard University, Cambridge, MA, USA, Tech. Rep. TR-10-98, Aug. 1998.

[27] K. Heafield, "KenLM: Faster and Smaller Language Model Queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom, July 2011, pp. 187–197.

[28] F. J. Och and H. Ney, "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Philadelphia, PA, USA, July 2002, pp. 295–302.

[29] C. Cherry and G. Foster, "Batch Tuning Strategies for Statistical Machine Translation," in *Proc. of the Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics (HLT-NAACL)*, Montreal, Canada, June 2012, pp. 427–436.

[30] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proc. of the 40th Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Philadelphia, PA, USA, July 2002, pp. 311–318.

[31] M. Freitag, J. Wuebker, S. Peitz, H. Ney, M. Huck, A. Birch, N. Durrani, P. Koehn, M. Mediani, I. Slawik, J. Niehues, E. Cho, A. Waibel, N. Bertoldi, M. Cettolo, and M. Federico, "Combined Spoken Language Translation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, South Lake Tahoe, CA, USA, Dec. 2014.

[32] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, MD, USA, June 2014.

[33] N. Durrani, A. Fraser, H. Schmid, H. Hoang, and P. Koehn, "Can markov models over minimal translation units help phrase-based smt?" in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 399–405. [Online]. Available: http://www.aclweb.org/anthology/P13-2071

[34] N. Durrani, A. Fraser, and H. Schmid, "Model With Minimal Translation Units, But Decode With Phrases," in *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, June 2013, pp. 1–11.

[35] D. Chiang, K. Knight, and W. Wang, "11,001 new features for statistical machine translation," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, June 2009, pp. 218–226.

[36] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, November 2006.

[37] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks." in *EMNLP*, 2013, pp. 1044–1054.

[38] J. Gao, X. He, W.-t. Yih, and L. Deng, "Learning continuous phrase representations for translation modeling," in *Proc. ACL*, 2014.

[39] A. Vaswani, Y. Zhao, V. Fossum, and D. Chiang, "Decoding with Large-Scale Neural Language Models Improves Translation," in *EMNLP*, 2013, pp. 1387–1392.

[40] P. Baltescu, P. Blunsom, and H. Hoang, "OxLM: A Neural Language Modelling Framework for Machine Translation," *The Prague Bulletin of Mathematical Linguistics*, vol. 102, no. 1, pp. 81–92, 2014.

[41] N. Durrani, H. Sajjad, H. Hoang, and P. Koehn, "Integrating an Unsupervised Transliteration Model into Statistical Machine Translation," in *Proceedings of the 15th Conference of the European Chapter of the ACL (EACL 2014)*. Gothenburg, Sweden: Association for Computational Linguistics, April 2014.

[42] H. Sajjad, A. Fraser, and H. Schmid, "A Statistical Model for Unsupervised and Semi-supervised Transliteration Mining," in *ACL12*, Jeju, Korea, 2012.

[43] N. Habash and F. Sadat, "Arabic Preprocessing Schemes for Statistical Machine Translation," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, New York City, USA, June 2006, pp. 49–52.

[44] A. Axelrod, X. He, and J. Gao, "Domain Adaptation via Pseudo In-Domain Data Selection," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK, July 2011, pp. 355–362.

[45] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes," in *Ninth Conference the European Chapter of the Association for Computational Linguistics (EACL)*, June 1999, pp. 71–76.

[46] M. Collins, P. Koehn, and I. Kucerova, "Clause Restructuring for Statistical Machine Translation," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, June 2005, pp. 531–540.

[47] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL)*, 2003.

[48] H. Schmid, "Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors," in *Proc. of the Int. Conf. on Computational Linguistics (COLING)*, Geneva, Switzerland, Aug. 2004.

[49] A. Bisazza, N. Ruiz, and M. Federico, "Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, USA, Dec. 2011, pp. 136–143.

[50] M. Huck, H. Hoang, and P. Koehn, "Augmenting String-to-Tree and Tree-to-String Translation with Non-Syntactic Phrases," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Baltimore, MD, USA, June 2014, pp. 486–498.

[51] ——, "Preference Grammars and Soft Syntactic Constraints for GHKM Syntax-based Statistical Machine Translation," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, Oct. 2014, pp. 148–156.

[52] M. Freitag, M. Huck, and H. Ney, "Jane: Open Source Machine Translation System Combination," in *Proc. of the Conf. of the European Chapter of the Assoc. for Computational Linguistics (EACL)*, Gothenburg, Sweden, Apr. 2014, pp. 29–32.

Combined Spoken Language Translation

[*]Markus Freitag, [*]Joern Wuebker, [*]Stephan Peitz, [*]Hermann Ney,
[‡]Matthias Huck, [‡]Alexandra Birch, [‡]Nadir Durrani, [‡]Philipp Koehn,
[†]Mohammed Mediani, [†]Isabel Slawik, [†]Jan Niehues, [†]Eunah Cho, [†]Alex Waibel,
[§]Nicola Bertoldi, [§]Mauro Cettolo, [§]Marcello Federico
[*]RWTH Aachen University, Aachen, Germany
[‡]University of Edinburgh, Edinburgh, Scotland
[†]Karlsruhe Institute of Technology, Karlsruhe, Germany
[§]Fondazione Bruno Kessler, Trento, Italy
[*]{freitag,wuebker,peitz,ney}@cs.rwth-aachen.de
[‡]a.birch@ed.ac.uk {mhuck,dnadir,pkoehn}@inf.ed.ac.uk
[†]{firstname.lastname}@kit.edu
[§]{bertoldi,cettolo,federico}@fbk.eu

Abstract

EU-BRIDGE[1] is a European research project which is aimed at developing innovative speech translation technology. One of the collaborative efforts within EU-BRIDGE is to produce joint submissions of up to four different partners to the evaluation campaign at the 2014 International Workshop on Spoken Language Translation (IWSLT). We submitted combined translations to the German→English spoken language translation (SLT) track as well as to the German→English, English→German and English→French machine translation (MT) tracks. In this paper, we present the techniques which were applied by the different individual translation systems of RWTH Aachen University, the University of Edinburgh, Karlsruhe Institute of Technology, and Fondazione Bruno Kessler. We then show the combination approach developed at RWTH Aachen University which combined the individual systems. The consensus translations yield empirical gains of up to 2.3 points in BLEU and 1.2 points in TER compared to the best individual system.

1. Introduction

The EU-BRIDGE project is funded by the European Union under the Seventh Framework Programme (FP7) and brings together several project partners who have each previously been very successful in contributing to advancements in automatic speech recognition and statistical machine translation. A number of languages and language pairs (both well-covered and under-resourced ones) are tackled with automatic speech recognition (ASR) and MT technology with different use cases in mind. Four of the EU-BRIDGE project partners are particularly experienced in machine translation for European language pairs: RWTH Aachen University (RWTH), the University of Edinburgh (UEDIN), Karlsruhe Institute of Technology (KIT), and Fondazione Bruno Kessler (FBK) have all regularly participated in large-scale evaluation campaigns like IWSLT and WMT in recent years, thereby demonstrating their ability to continuously enhance their systems and promoting progress in machine translation. Machine translation research within EU-BRIDGE has a strong focus on translation of spoken language. The IWSLT TED talks task constitutes an interesting framework for empirical testing of some of the systems for spoken language translation which are developed as part of the project.

In this work, we describe the EU-BRIDGE submissions to the 2014 IWSLT translation task. This year, we combined several single systems of RWTH, UEDIN, KIT, and FBK for the German→English SLT, German→English MT, English→German MT, and English→French MT tasks. Additionally to the standard system combination pipeline presented in [1, 2], we applied a recurrent neural network rescoring step [3] for the English→French MT task. Similar cooperative approaches based on system combination have proven to be valuable for machine translation in previous joint submissions, e.g. [4, 5].

2. RWTH Aachen University

RWTH applied the identical training pipeline and models on both language pairs: The state-of-the-art phrase-based baseline systems were augmented with a hierarchical reordering model, several additional language models (LMs) and maximum expected BLEU training for phrasal, lexical and reordering models. Further, RWTH employed rescoring with novel recurrent neural language and translation models. The same systems were used for the SLT track, where RWTH ad-

[1] http://www.eu-bridge.eu

ditionally performed punctuation prediction on the automatic transcriptions employing hierarchical phrase-based translation. Both the phrase-based and the hierarchical decoder are implemented in RWTH's publicly available translation toolkit Jane [6, 7]. The model weights of all systems were tuned with standard Minimum Error Rate Training [8] on the provided dev2012 set. RWTH used BLEU as optimization objective. For the German→English translation direction, in a preprocessing step the German source was decompounded [9] and part-of-speech-based long-range verb reordering rules [10] were applied. RWTH's translation systems are described in more detail in [11].

Backoff Language Models
Each translation system used three backoff LMs that were estimated with the KenLM toolkit [12]: A large general domain 5-gram LM, an in-domain 5-gram LM and a 7-gram word class language model (wcLM). All of them used interpolated Kneser-Ney smoothing. For the general domain LM, RWTH first selected $\frac{1}{2}$ of the English Shuffled News, and $\frac{1}{4}$ of the French Shuffled News as well as both the English and French Gigaword corpora by the cross-entropy difference criterion described in [13]. The selection was then concatenated with all available remaining monolingual data and used to build and unpruned LM. The in-domain language models were estimated on the TED data only. For the word class LM, RWTH trained 200 classes on the target side of the bilingual training data using an in-house tool similar to `mkcls` [14]. With these class definitions, RWTH applied the technique shown in [15] to compute the wcLM on the same data as the general-domain LM.

Maximum Expected BLEU Training
RWTH applied discriminative training, learning three types of features under a maximum expected BLEU objective [16]. It was performed on the TED portion of the data, which is high quality in-domain data of reasonable size. This makes training feasible while at the same time providing an implicit domain adaptation effect. Similar to [16], RWTH generated 100-best lists on the training data which were used as training samples for a gradient based update method. Leave-one-out [17] was applied to circumvent over-fitting. Here, RWTH followed an approach similar to [18], where each feature type was condensed into a single feature for the log-linear model combination. In the first pass, RWTH trained phrase pair and phrase-internal word pair features, and in the second pass a hierarchical reordering model, resulting altogether in an additional eight models for log-linear combination.

Recurrent Neural Network Models
All systems applied rescoring on 1000-best lists using recurrent language and translation models. The recurrency was handled with the long short-term memory (LSTM) architecture [19] and RWTH used a class-factored output layer for increased efficiency as described in [20]. All neural networks were trained on the TED portion of the data with 2000 word classes. In addition to the recurrent language model (RNN-LM), RWTH applied the deep bidirectional word-based translation model (RNN-BTM) described in [3], which is capable of taking the *full source context* into account for each translation decision.

Spoken Language Translation
For the SLT task, RWTH reintroduced punctuation and case information before the actual translation similar to [21]. However, RWTH employed a hierarchical phrase-based system with a maximum of one nonterminal symbol per rule in place of a phrase-based system. A punctuation prediction system based on hierarchical translation is able to capture long-range dependencies between words and punctuation marks and is more robust for unseen word sequences. The model weights are tuned with standard MERT on 100-best lists. As optimization criterion RWTH used F_2-Score rather than BLEU or WER. More details can be found in [22].

Since punctuation predicting and recasing were applied before the actual translation, the final translation systems from the MT track could be kept completely unchanged.

3. University of Edinburgh

The UEDIN translation engines [23] are based on the open source Moses toolkit [24]. UEDIN set up phrase-based systems for all SLT and MT tasks covered in this paper, and additionally a string-to-tree syntax-based system [25] for the English→German MT task. The systems were trained using monolingual and parallel data from WIT^3, Europarl, MultiUN, the English and French Gigaword corpora as provided by the Linguistic Data Consortium, the German Political Speeches Corpus, and the Common Crawl, 10^9, and News Commentary corpora from the WMT shared task training data. Word alignments for the MT track systems were created by aligning the data in both directions with MGIZA++ [26] and symmetrizing the two trained alignments. Word alignments for the SLT track system were created using fast_align [27]. The SRILM toolkit [28] was employed to train 5-gram LMs with modified Kneser-Ney smoothing [29]. UEDIN trained individual LMs on each corpus and then interpolated them using weights tuned to minimize perplexity on a development set.

Common features included in the UEDIN phrase-based systems are the language model, phrase translation scores in both directions smoothed with Good-Turing discounting, lexical translation scores in both directions, word and phrase penalties, six simple count-based binary features, distance-based distortion costs, a hierarchical lexicalized reordering model [30], sparse lexical and domain indicator features [31] and operation sequence models over different word representations [32]. Model weights were optimized with batch MIRA [33] to maximize BLEU [34].

Spoken Language Translation
One of the main challenges of spoken language translation is to overcome the mismatch in the style of data that the

speech recognition systems output, and the written text that is used to train the translation model. ASR system output lacks punctuation and capitalization, which is the main stylistic differences. Previous research [35, 21, 36] suggests that it is preferrable to punctuate the text before translation, which is what UEDIN did by training a translation system on the German side of the parallel data. The "source language" of the system had punctuation and capitalization stripped, and the "target language" was the standard German parallel text. The handling of punctuation is similar to the other groups in this paper, however UEDIN used a phrase-based model with no distortion or reordering, and tuned the model to the ASR input text using batch MIRA and the BLEU score.

German→English MT
For the UEDIN German→English MT task system, pre-reordering [37] and compound splitting [38] were applied to the German source language side in a preprocessing step. A factored translation model [39] was employed. Source side factors are word, lemma, part-of-speech (POS) tag, and morphological tag. Target side factors are word, lemma, and POS tag. UEDIN incorporated two additional LMs into the German→English MT system: a 7-gram LM over POS tags (trained on WIT^3 only) and a 7-gram LM over lemmas (trained on WIT^3 only). Model weights were optimized on a concatenation of dev2010 and dev2012.

English→French MT
UEDIN contributed two phrase-based systems for the English→French EU-BRIDGE system combination. Both comprise Brown clusters with 200 classes as additional factors on source and target side. The system denoted as UEDIN-A was trained without the MultiUN and 10^9 corpora, the system denoted as UEDIN-B was trained with all corpora. An additional feature incorporated into the systems is an LM over Brown clusters (UEDIN-A: 7-gram, UEDIN-B: 5-gram). Model weights were optimized on dev2010.

English→German MT
UEDIN contributed two phrase-based systems (UEDIN-A and UEDIN-B) and a syntax-based system (UEDIN-C) for English→German MT.

Phrase-based systems. UEDIN-A and UEDIN-B employ factored models. Source side factors are word, POS tag, and Brown cluster (2000 classes). Target side factors are word, POS tag, Brown cluster (2000 classes), and morphological tag. UEDIN-A was trained with all corpora, whereas for UEDIN-B the parallel training data was restricted to the in-domain WIT^3 corpus. Additional features of the systems are: a 5-gram LM over Brown clusters, a 7-gram LM over morphological tags (UEDIN-A: trained on all data, UEDIN-B: trained on WIT^3 only), and a 7-gram LM over POS tags (UEDIN-A, not UEDIN-B). Model weights of UEDIN-B were optimized on dev2010, model weights of UEDIN-A on a concatenation of dev2010 and dev2012.

Syntax-based system. UEDIN-C is a string-to-tree translation system with similar features as the ones described in [40]. The target-side data was parsed with BitPar [41], and right binarization was applied to the parse trees. The system was adapted to the TED domain by extracting separate rule tables (from the WIT^3 corpus and from the rest of the parallel data) and merging them with a fill-up technique [42]. Augmenting the system with non-syntactic phrases [43] and adding soft source syntactic constraints [44] yielded further improvements. Model weights of UEDIN-C were optimized on a concatenation of dev2010 and dev2012.

4. Karlsruhe Institute of Technology

The KIT translations were generated by an in-house phrase-based translations system [45]. The models were trained on the Europarl, News Commentary, WIT^3, Common Crawl corpora for all directions, as well as on the additional monolingual training data. The noisy Crawl corpora were filtered using an SVM classifier [46]. In addition to the standard preprocessing, KIT used compound splitting [38] for the German text when translating from German. In the SLT task, KIT first recased the input and added punctuation marks to the ASR hypotheses. This was done with a monolingual translation system as shown in [36].

In all translation directions, KIT used a pre-reordering approach. Different reorderings of the source sentences were encoded in a word lattice. For the English→French system, only short-range rules were used to generate these lattices [47]. Long-range rules [48] and tree-based reordering rules [49] were used for German→English. The POS tags needed for these rules were generated by the TreeTagger [50] and the parse trees by the Stanford Parser [51]. In addition, for the language pairs involving German KIT applied the different reorderings of both language pairs using a lexicalized reordering model. The phrase tables of the systems were trained using GIZA++ alignment [52]. KIT adapted the phrase table to the TED domain using the backoff approach and by means of candidate selection [53]. In addition to the phrase table probabilities, KIT modeled the translation process by a bilingual language model [54] and a discriminative word lexicon using source context features [55].

During decoding, KIT used several LMs to adapt the system to the task and to better model the sentence structure using a class-based LM. For the German→English task, KIT used one LM trained on all data, an in-domain LM trained only on the WIT^3 corpus, and one LM trained on 5M sentences selected using cross-entropy difference [13]. As classes KIT used the clusters obtained using the `mkcls` algorithm on the WIT^3 corpus. For German↔ English, KIT used a 9-gram LM with 100 or 1000 clusters and for the English→French MT task, a cluster-based 4-gram LM was trained on 500 clusters. For English→German, KIT also used a 9-gram POS-based LM. The log-linear combination of all these models was optimized on the provided development data using MERT.

5. Fondazione Bruno Kessler

The FBK system was built upon a standard phrase-based system using the Moses toolkit [24], and exploited the huge amount of parallel English-French and monolingual French training data provided by the organizers. It featured a statistical log-linear model including a phrase-based translation model (TM) and lexicalized phrase-based reordering models (RM), two French language models (LMs), as well as distortion, word and phrase penalties. Tuning of the system was performed on dev2010 by optimizing BLEU using Minimum Error Rate Training [8]. It is worth noticing that all available development data sets, namely dev2010 and test2010-2012, were added to the in-domain training data to build the system actually employed for the 2014 evaluation campaign.

In order to adapt the system on TED specific domain and genre and to reduce the size of the system, data selection was carried out on all parallel English-French corpora, using the whole WIT^3 [56] training corpus as in-domain data. Data selection was performed by means of XenC toolkit [57] exploiting bilingual cross-entropy difference [58] separately for each available training corpus except the in-domain WIT^3 data. Different amount of texts were selected from each corpora ranging from 2% to 30%, and then concatenating for building one parallel corpus containing 2.6M sentences for a total of 57M English and 63M French running words.

Two TMs and two RMs were trained independently on the parallel in-domain and selected data, using the standard Moses procedure and MGIZA++ toolkit [26] for word-alignment; TMs and RMs were combined using the back-off technique (for both TM and RM), taking WIT^3 as primary component, for a total of 168M phrase pairs. The back-off table combination is similar to the fill-up technique [42], but does not add any provenance binary features.

The French side of in-domain and selected data were also employed to estimate a 2-component mixture language model [59]. Moreover, a second huge French LM was estimated on all permitted monolingual French data consisting of $\sim$1.4G running words, as a mixture of 8 components. Both LMs have order 5 and were smoothed by means of the interpolated Improved Kneser-Ney method [29]; they include 57M and 661M 5-grams, respectively. A full description of the system can be found in the FBK system paper.

6. System Combination

In this section, we give a brief re-introduction of confusion network system combination. System combination is used to produce consensus translations from multiple hypotheses which are outputs of different translation engines. The consensus translations can be better in terms of translation quality than any of the individual hypotheses. To combine the engines of the project partners for the EU-BRIDGE joint setups, we applied a system combination implementation that has been developed at RWTH Aachen University [1].

In Fig. 1 an overview is illustrated. We first address the generation of a confusion network (CN) from I input translations. For that we need a pairwise alignment between all input hypotheses. This alignment is calculated via METEOR [60]. The hypotheses are then reordered to match the word order of a selected skeleton hypothesis. Instead of using only one of the input hypothesis as skeleton, we generate I different CNs, each having one of the input systems as skeleton. The final lattice is the union of all I previous generated CNs. In Fig. 2 an example confusion network of $I = 4$ input translations with one skeleton translation is illustrated. Between two adjacent nodes, we always have a choice between the I different system output words. The confusion network decoding step involves determining the shortest path through the network. Each arc is assigned one score which is a linear model combination (Eq. 1) of M different models.

$$\sum_{m=1}^{M} \lambda_m h_m \tag{1}$$

The standard set of models is a word penalty, a 3-gram language model trained on the input hypotheses, and for each system one binary voting feature. During decoding the binary voting feature for system i ($1 \leq i \leq I$) is 1 iff the word is from system i, otherwise 0. The M different model weights λ_m are trained with MERT [8].

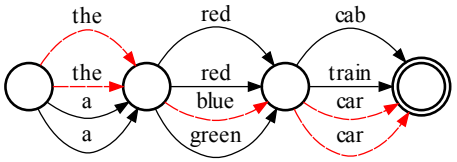


Figure 2: System A: *the red cab* ; System B: *the red train* ; System C: *a blue car* ; System D: *a green car* ; Reference: *the blue car* .

7. Results

In this section, we present our experimental results. All reported BLEU [34] and TER [61] scores are case-sensitive with one reference. All system combination results have been generated with RWTH's open source system combination implementation Jane [1].

German→English SLT

For the German→English SLT task, we combined three different individual systems generated by UEDIN, KIT, and RWTH. Experimental results are given in Table 1. The final system combination yields improvements of 1.5 points in BLEU and 1.2 points in TER compared to the best single system (KIT). All single systems as well as the system combination parameters were tuned on dev2012. For this year's IWSLT SLT track, dev2012 was the only given test set containing automatic speech recognition output.

German→English MT

Similar to the SLT track, the German→English MT system combination submission is a combined translation of three different individual systems by UEDIN, KIT, and RWTH.

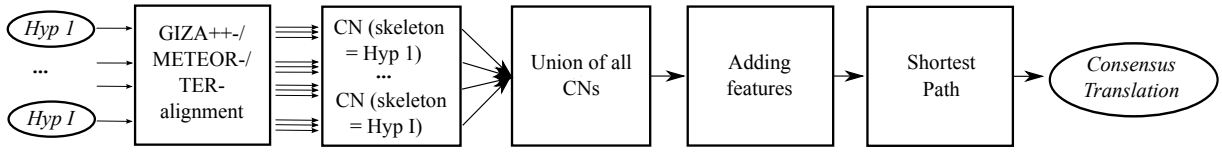


Figure 1: Confusion network decoding structure.

Table 1: Results for the German→English SLT task.

system	dev2012	
	BLEU	TER
KIT	20.7	60.5
RWTH	20.8	61.4
UEDIN	20.3	63.0
syscom	22.2	59.3

Table 2: Results for the German→English MT task.

system	tst2010		tst2011		tst2012	
	BLEU	TER	BLEU	TER	BLEU	TER
KIT	31.5	47.6	37.1	42.5	32.0	47.6
RWTH	31.8	47.2	38.3	41.3	32.0	47.0
UEDIN	31.6	47.6	37.3	42.5	31.7	47.9
syscom	33.3	46.1	39.4	40.6	33.5	46.2

Experimental results are given in Table 2. The system combination parameters have been optimized on test2012. Compared to the best individual system (RWTH), the system combination improved translation scores by up to 1.5 points in BLEU and 1.1 points in TER.

English→French MT

For the English→French MT task, we combined five different individual systems. FBK, KIT, and RWTH provided one individual system output for the system combination. UEDIN added one advanced contrastive system in addition to their primary system. Experimental results are given in Table 3. The system combination of all five individual systems yields an improvement of up to 0.6 points in BLEU compared to the best RWTH individual system output. Using a recurrent neural network (RNN) LM to rescore a 1000-best list of the system combination output, leads to a small translation improvement of +0.1 in BLEU. The same RNN LM was applied in the best individual system of RWTH Aachen. The improvements are only small, as the model is already contained the best individual system.

English→German MT

For the English→German setup, we combined three different individual system setups of UEDIN with the primary submission of KIT. Experimental results are given in Table 4. All system combination parameters are tuned on tst2012. The EU-BRIDGE submission enhanced the translation quality by up to 1.4 points in BLEU and 1.2 points in TER compared to the best individual system.

Table 3: Results for the English→French MT task.

system	tst2010		tst2011		tst2012	
	BLEU	TER	BLEU	TER	BLEU	TER
FBK	32.8	50.4	39.2	42.6	40.0	41.4
KIT	33.1	48.4	37.3	42.5	39.1	40.2
RWTH	34.5	47.6	41.1	40.1	42.0	38.6
UEDIN-A	33.6	48.5	40.2	40.6	41.0	39.6
UEDIN-B	33.2	49.1	39.1	42.0	40.7	39.8
syscom	35.1	48.5	41.7	41.4	44.0	38.7
+RNN	35.2	48.5	41.7	41.3	44.3	38.5

Table 4: Results for the English→German MT task.

system	tst2010		tst2011		tst2012	
	BLEU	TER	BLEU	TER	BLEU	TER
KIT	24.5	55.2	27.1	50.5	23.5	56.0
UEDIN-A	24.9	55.5	27.8	50.1	23.4	56.9
UEDIN-B	24.1	55.7	26.7	50.8	22.2	57.3
UEDIN-C	24.8	55.3	26.5	50.5	23.1	56.6
syscom	25.9	54.0	28.1	49.1	24.9	55.0

8. Conclusion

We achieved better translation performance with gains of up to +2.3 points in BLEU and -1.2 points in TER by combining the different system hypotheses of up to four partners of the EU-BRIDGE project. The four research institutes (RWTH Aachen University, University of Edinburgh, Karlsruhe Institute of Technology, Fondazione Bruno Kessler) are maintaining different machine translation engines based on different approaches. System combination combined all the different advancements of all engines together into our final submissions. For English→French we applied a recurrent neural network language model in an additional rescoring step which only gives small improvement of +0.1 points in BLEU.

9. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

10. References

[1] M. Freitag, M. Huck, and H. Ney, "Jane: Open Source Machine Translation System Combination," in *Proc. of the Conf. of the European Chapter of the Assoc. for Computational Linguistics (EACL)*, Gothenburg, Sweden, Apr. 2014, pp. 29–32.

[2] E. Matusov, N. Ueffing, and H. Ney, "Computing Con-

sensus Translation from Multiple Machine Translation Systems Using Enhanced Hypotheses Alignment," in *Proc. of the Conf. of the European Chapter of the Assoc. for Computational Linguistics (EACL)*, Trento, Italy, Apr. 2006, pp. 33–40.

[3] M. Sundermeyer, T. Alkhouli, J. Wuebker, and H. Ney, "Translation modeling with bidirectional recurrent neural networks," in *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014.

[4] M. Freitag, S. Peitz, J. Wuebker, H. Ney, M. Huck, R. Sennrich, N. Durrani, M. Nadejde, P. Williams, P. Koehn, T. Herrmann, E. Cho, and A. Waibel, "EU-BRIDGE MT: Combined Machine Translation," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Baltimore, MD, USA, June 2014, pp. 105–113.

[5] M. Freitag, S. Peitz, J. Wuebker, H. Ney, N. Durrani, M. Huck, P. Koehn, T.-L. Ha, J. Niehues, M. Mediani, T. Herrmann, A. Waibel, N. Bertoldi, M. Cettolo, and M. Federico, "EU-BRIDGE MT: Text Translation of Talks in the EU-BRIDGE Project," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, Dec. 2013, pp. 128–135.

[6] D. Vilar, D. Stein, M. Huck, and H. Ney, "Jane: Open source hierarchical translation, extended with reordering and lexicon models," in *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July 2010, pp. 262–270.

[7] J. Wuebker, M. Huck, S. Peitz, M. Nuhn, M. Freitag, J.-T. Peter, S. Mansour, and H. Ney, "Jane 2: Open Source Phrase-based and Hierarchical Statistical Machine Translation," in *COLING '12: The 24th Int. Conf. on Computational Linguistics*, Mumbai, India, Dec. 2012, pp. 483–491.

[8] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003, pp. 160–167.

[9] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *Proceedings of European Chapter of the ACL (EACL 2009)*, 2003, pp. 187–194.

[10] M. Popović and H. Ney, "POS-based Word Reorderings for Statistical Machine Translation," in *International Conference on Language Resources and Evaluation*, 2006, pp. 1278–1283.

[11] J. Wuebker, S. Peitz, A. Guta, and H. Ney, "The RWTH Aachen Machine Translation Systems for IWSLT 2014," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, South Lake Tahoe, CA, USA, Dec. 2014.

[12] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable modified Kneser-Ney language model estimation," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Sofia, Bulgaria, August 2013, pp. 690–696.

[13] R. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Uppsala, Sweden, July 2010, pp. 220–224.

[14] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes." in *Proc. of the Conf. of the European Chapter of the Assoc. for Computational Linguistics (EACL)*, 1999.

[15] J. Wuebker, S. Peitz, F. Rietig, and H. Ney, "Improving Statistical Machine Translation with Word Class Models," in *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, Seattle, WA, USA, Oct. 2013, pp. 1377–1381.

[16] X. He and L. Deng, "Maximum Expected BLEU Training of Phrase and Lexicon Translation Models," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Jeju, Republic of Korea, July 2012, pp. 292–301.

[17] J. Wuebker, A. Mauser, and H. Ney, "Training Phrase Translation Models with Leaving-One-Out," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Uppsala, Sweden, July 2010, pp. 475–484.

[18] M. Auli, M. Galley, and J. Gao, "Large Scale Expected BLEU Training of Phrase-based Reordering Models," in *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[20] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Interspeech*, Portland, OR, USA, Sept. 2012.

[21] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling Punctuation Prediction as Machine Translation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, USA, Dec. 2011.

[22] S. Peitz, M. Freitag, and H. Ney, "Better Punctuation Prediction with Hierarchical Phrase-Based Translation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, South Lake Tahoe, CA, USA, Dec. 2014.

[23] A. Birch, M. Huck, N. Durrani, N. Bogoychev, and P. Koehn, "Edinburgh SLT and MT System Description for the IWSLT 2014 Evaluation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, South Lake Tahoe, CA, USA, Dec. 2014.

[24] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Prague, Czech Republic, June 2007, pp. 177–180.

[25] M. Galley, M. Hopkins, K. Knight, and D. Marcu, "What's in a translation rule?" in *Proc. of the Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics (HLT-NAACL)*, Boston, MA, USA, May 2004, pp. 273–280.

[26] Q. Gao and S. Vogel, "Parallel Implementations of Word Alignment Tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, ser. SETQA-NLP '08, Columbus, OH, USA, June 2008, pp. 49–57.

[27] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," in *Proc. of the Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics (HLT-NAACL)*, Atlanta, GA, USA, June 2013, pp. 644–648.

[28] A. Stolcke, "SRILM – an Extensible Language Modeling Toolkit," in *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP)*, Denver, CO, USA, Sept. 2002, pp. 901–904.

[29] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," Computer Science Group, Harvard University, Cambridge, MA, USA, Tech. Rep. TR-10-98, Aug. 1998.

[30] M. Galley and C. D. Manning, "A Simple and Effective Hierarchical Phrase Reordering Model," in *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, Honolulu, HI, USA, Oct. 2008, pp. 847–855.

[31] E. Hasler, B. Haddow, and P. Koehn, "Sparse Lexicalised Features and Topic Adaptation for SMT," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, Dec. 2012, pp. 268–275.

[32] N. Durrani, P. Koehn, H. Schmid, and A. Fraser, "Investigating the Usefulness of Generalized Word Representations in SMT," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland, Aug. 2014, pp. 421–432.

[33] C. Cherry and G. Foster, "Batch Tuning Strategies for Statistical Machine Translation," in *Proc. of the Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics (HLT-NAACL)*, Montréal, Canada, June 2012, pp. 427–436.

[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Philadelphia, PA, USA, July 2002, pp. 311–318.

[35] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, November 2006.

[36] E. Cho, J. Niehues, and A. Waibel, "Segmentation and Punctuation Prediction in Speech Language Translation using a Monolingual Translation System," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[37] M. Collins, P. Koehn, and I. Kucerova, "Clause Restructuring for Statistical Machine Translation," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Ann Arbor, MI, USA, June 2005, pp. 531–540.

[38] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *Proc. of the Conf. of the European Chapter of the Assoc. for Computational Linguistics (EACL)*, Budapest, Hungary, Apr. 2003, pp. 187–194.

[39] P. Koehn and B. Haddow, "Interpolated Backoff for Factored Translation Models," in *Proc. of the Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, San Diego, CA, USA, Oct./Nov. 2012.

[40] P. Williams, R. Sennrich, M. Nadejde, M. Huck, E. Hasler, and P. Koehn, "Edinburgh's Syntax-Based Systems at WMT 2014," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Baltimore, MD, USA, June 2014, pp. 207–214.

[41] H. Schmid, "Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors," in *Proc. of the Int. Conf. on Computational Linguistics (COLING)*, Geneva, Switzerland, Aug. 2004.

[42] A. Bisazza, N. Ruiz, and M. Federico, "Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, USA, Dec. 2011, pp. 136–143.

[43] M. Huck, H. Hoang, and P. Koehn, "Augmenting String-to-Tree and Tree-to-String Translation with Non-Syntactic Phrases," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Baltimore, MD, USA, June 2014, pp. 486–498.

[44] Huck, Matthias and Hoang, Hieu and Koehn, Philipp, "Preference Grammars and Soft Syntactic Constraints for GHKM Syntax-based Statistical Machine Translation," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, Oct. 2014, pp. 148–156.

[45] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *International Conference on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[46] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The KIT English-French Translation systems for IWSLT 2011," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, USA, 2011.

[47] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Skövde, Sweden, 2007.

[48] J. Niehues, T. Herrmann, M. Kolss, and A. Waibel, "The Universität Karlsruhe Translation System for the EACL-WMT 2009," in *Proceedings of the Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece, 2009.

[49] T. Herrmann, J. Niehues, and A. Waibel, "Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation," in *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, Altanta, GA, USA, June 2013.

[50] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *Int. Conf. on New Methods in Language Processing*, Manchester, UK, 1994.

[51] A. N. Rafferty and C. D. Manning, "Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines," in *Proc. of the Workshop on Parsing German*, Columbus, OH, USA, 2008.

[52] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[53] J. Niehues and A. Waibel, "Detailed Analysis of Different Strategies for Phrase Table Adaptation in SMT," in *Proc. of the Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, 2012.

[54] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, Edinburgh, UK, 2011.

[55] J. Niehues and A. Waibel, "An MT Error-Driven Discriminative Word Lexicon using Sentence Structure Features," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, 2013, pp. 512–520.

[56] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proc. of the Annual Conf. of the European Assoc. for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[57] A. Rousseau, "Xenc: An open-source tool for data selection in natural language processing," *The Prague Bulletin of Mathematical Linguistics*, no. 100, pp. 73–82, 2013.

[58] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, 2011, pp. 355–362.

[59] M. Federico and R. De Mori, "Language modelling," in *Spoken Dialogues with Computers*, R. D. Mori, Ed. London, UK: Academy Press, 1998, ch. 7, pp. 199–230.

[60] S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments," in *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, Ann Arbor, MI, USA, June 2005, pp. 65–72.

[61] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proc. of the Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*, Cambridge, MA, USA, Aug. 2006, pp. 223–231.

The MITLL-AFRL IWSLT 2014 MT System[†]

Michaeel Kazi, Elizabeth Salesky, Brian Thompson, Jessica Ray, Michael Coury, Wade Shen

MIT Lincoln Laboratory
Human Language Technology Group
244 Wood Street
Lexington, MA 02420, USA
{michaeel.kazi, elizabeth.salesky, brian.thompson, jessica.ray, michael.coury, swade}@ll.mit.edu

Tim Anderson, Grant Erdmann, Jeremy Gwinnup, Katherine Young, Brian Ore, Michael Hutt

Air Force Research Laboratory
Human Effectiveness Directorate
2255 H Street
Wright-Patterson AFB, OH 45433
{timothy.anderson.20, grant.erdmann, jeremy.gwinnup.ctr, katherine.young.1.ctr, brian.ore.ctr, michael.hutt.ctr}@us.af.mil

Abstract

This report summarizes the MITLL-AFRL MT and ASR systems and the experiments run using them during the 2014 IWSLT evaluation campaign. Our MT system is much improved over last year, owing to integration of techniques such as PRO and DREM optimization, factored language models, neural network joint model rescoring, multiple phrase tables, and development set creation. We focused our efforts this year on the tasks of translating from Arabic, Russian, Chinese, and Farsi into English, as well as translating from English to French. ASR performance also improved, partly due to increased efforts with deep neural networks for hybrid and tandem systems. Work focused on both the English and Italian ASR tasks.

1. Introduction

During the evaluation campaign for the 2014 International Workshop on Spoken Language Translation (IWSLT'14) [1] our experimental efforts in machine translation (MT) centered on 1) decoding with factored language models [2], 2) neural network joint model [3] rescoring, 3) multiple phrase tables, and 4) development set creation. Other algorithms in our toolbox included the recurrent neural network language model [4], and the operational sequence models [5].

Experimental efforts for the automatic speech recognition (ASR) task focused on the use of deep neural networks for use in both hybrid and tandem configurations. Updated language models also improved performance compared to our 2013 system.

[†]This work is sponsored by the Air Force Research Laboratory under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

We here describe improvements over our 2013 submission systems. For a more in-depth description of the 2013 system, refer to [6]. This paper is structured as follows. Section 2 presents our work on the MT task, and discusses each of the techniques mentioned above, ending with a discussion of submitted systems. Our work on the ASR task is discussed in Section 3.

2. Machine Translation

2.1. Data usage

Unless otherwise noted, data described in this section originates from the WMT14 website[1]. We used the in-domain data supplied by WIT3 [7] for all language pairs. In English-French, our parallel data included the 10^9 corpus, News Commentary v8, Europarl v7, and the UN corpus. In Russian to English, we used the Yandex corpus[2], Common Crawl, Wiki Headlines, News Crawl, and UN data. In Arabic to English, we used only the UN data, which was sentence-aligned via Champollion [8].

Extra monolingual data (in addition to parallel data) included the News Crawl corpus 2007-2011 (English and French), LDC Gigaword English v5 [9], and LDC French Gigaword v3 [10].

2.2. Data Preprocessing and Cleanup

The TED datasets were examined for repetition errors, in which English sentences or sentence-internal phrases are translated multiple times. These errors derive from the TED website. When repetition errors occur in training data, they cause alignment problems; when they occur in test data, they degrade the machine translation.

[1] http://www.statmt.org/wmt14/translation-task.html
[2] https://translate.yandex.ru/corpus?lang=en

Repeated phrases of more than 10 words were detected and removed. If parallel text was available, phrases were only removed when there was no corresponding repetition in the English sentence. The Farsi test sets contained substantial repetition; lesser amounts were found in the Chinese dev and test data, and in the French dev data. Arabic and Russian dev and test sets were also examined, but did not contain these repetitions. Removing the repetitions from the Farsi `tst2014` file improved BLEU +1.53, based on last year's IWSLT system. We expect to see some improvement for Chinese as well, but due to time constraints defer that comparison to future work. Repeat statistics for the dev and test sets are outlined in Table 1, and for the train sets in Table 2.

Lang.	Set	Repeats	Length
French	`dev2010`	11	887
Chinese	`dev2010`	87	887
	`tst2010`	81	1570
	`tst2014`	13	1068
Farsi	`tst2010`	1	885
	`tst2011`	22	1132
	`tst2012`	343	1375
	`tst2013`	187	923
	`tst2014`	53	1131

Table 1: Repeated sentences per dev/test set

Lang.	Year	Repeats	Length
Arabic	2013	3	155,047
	2014	5	186,467
Chinese	2014	550	177,901
Farsi	2013	5,749	81,872
	2014	8,987	112,704
French	2013	173	162,681
	2014	373	186,510
Russian	2013	109	135,669
	2014	145	185,205

Table 2: Repeated sentences per training set

2.3. Baseline MT System

Our system implements a fairly standard phrase-based SMT [11] architecture. It consists of the following:

- Training corpora filtered for maximum sentence length of 40.
- MADAMIRA Beta 1.0 [12] tokenization for Arabic, Stanford Segmenter [13] + character segmentation for Chinese, Moses tokenizer for Russian and English.
- GIZA++ word alignments, using 100 word classes, Models 2-4 + HMM and optionally Model 5.
- Order 6 TED language model.
- Maximum extracted phrase length of 9.
- Monotone-at-punctuation, drop-unknown.
- Phrasetable with KN smoothing [14].
- Word-based [15] or hierarchical [16] monotone-swap-distort lexical reordering.
- Moses decoder [17], no reordering over punctuation, n-best list size 200.
- Rescore n-best-lists using order-7 class-based TED LM. Default is 80 word classes.
- Pairwise rank optimization [18] or Derivative-Free Robust Error Minimization (DREM) [6] over cumulative n-best lists.
- One-best result (we saw no consistent benefit to using Minimum Bayes Risk).

In addition to the tokenizers listed above, in English-French and the English component of the Arabic task, we used simple in-house tokenizers that separate out punctuation and common language specific constructions (e.g. l' in French). Reported scores are case-sensitive BLEU scores with separated punctuation (via MTEval[3]). To account for variance, unless otherwise stated, scores are averages over 10 optimizations. Baseline systems are tuned on `dev2010`.

2.3.1. Language Modeling

Language models on in-domain or target-side parallel data were trained using either MITLM [19] or SRILM 1.7 [20]. With the Gigaword dataset, we typically used `lmplz` [21]. All LMs were binarized using KenLM [22]. Word classes were trained using `mkcls` [23].

2.4. Additional Phrase Table Training

The use of extra phrase table training data was indispensible in the English to French and Russian to English tasks. For each of these, we used Moore-Lewis [24] cross-entropy filtering cE) and kept 10% of the out-of-domain data. We also experimented with a 2nd phrase table in Arabic to English and Russian to English using the MultiUN and Yandex datasets, respectively. These were tested in addition to a cross-entropy filtered PT.

Lang.	Baseline	cE PT	+ 2nd PT	+Backoff PT
en-fr	38.25†	41.39†	–	–
ar-en	30.94	31.55	31.53	30.66
ru-en	21.13	22.47	22.15	21.25

Table 3: Comparison of mean BLEU on `tst2013` with additional PT training. (†=`tst2012`)

2.5. Neural Network Joint Model

We replicated the architecture described in Devlin et al. Neural Network Joint Model [3], which is similar to a

[3] `http://www.itl.nist.gov/iad/mig/tests/mt/2009/`

continuous space language model, but conditioned on words in the source language as well. Each target side word is considered to be "affiliated" with a source word (via word alignments included in the phrase table). The affiliated word, the 5 words before and after it, and a 3-gram on the target side are input to the neural network; the outputs are posterior probabilities over the entire target language vocabulary.

We implemented this to rescore 200-best lists. Our results were promising; we saw modest gains on a variety of language pairs. Devlin et al. claim gains more than double when this is integrated into the decoder itself. This is future work for us.

We implemented the NNJM within Theano [25], and ran training and rescoring on a Tesla K40 GPU. We trained a vocabulary by taking words seen in TED 4 or more times. Additional words in the phrase table (such as from out-of-domain data) were mapped to word classes using `mkcls`. Training was done on the output of grow-diag-final-and alignments. In the case where out-of-domain data was available and useful, we first trained the network on only the out-of-domain data, then switched to in-domain data only. In building the phrase table, sub-phrase alignments for a given phrase pair were taken from the extracted phrase pair with maximum scoring lexical $p(f|e)$.

Lang.	RNNLM	NNJM-In	Out+In
ar-en	30.59	30.87	30.88
en-fr	40.85	39.75	41.39
ru-en	20.81	21.21	21.27

Table 4: Effects of neural joint model rescoring, mean BLEU over `tst2012`

2.6. Factored Models

Following the success of Edinburgh's Target Sequence Model [2] (and our own rescoring n-best lists via `mkcls`), we enabled factored language models within Moses. In theory this should be better than rescoring, because it will alter the search space the decoder traverses. For class-based LMs, we compared `mkcls` to Percy Liang's `brown-cluster`[4]. We saw that the optimal number of word classes varied, but once tuned, BLEU varied only 0.2% on ru-en. All numbers reported here use `mkcls`.

Lang.	Baseline	50	200	600	1000
ar-en	29.61	29.70	29.58	29.70	29.71
fa-en	16.68	16.26	16.54	16.62	16.87
ru-en	18.75	19.03	19.32	19.45	19.16
zh-en	15.06	14.95	14.64	14.80	15.00

Table 5: nClasses with factored LMs, `tst2013`.

[4] `https://github.com/percyliang/brown-cluster`

We saw further gains of 0.41, 0.37 and 1.2 with additional class-based Gigaword LMs in ar-en, fa-en, and ru-en, respectively. However, the results for zh-en were inconsistently bad. For instance, we saw a gain of 0.29 with 200 classes, a loss of 0.39 with 1000 classes, and all experiments were worse than the baseline score. Additional factored LMs, such as POS tags, were tried in Russian to English, but produced a loss in performance of 0.6 BLEU.

We also experimented with the operational sequence model over word classes. We saw significant gain in English to French using only TED data (+0.69 on `tst2010` using 100 classes), but using the full out-of-domain data, we did not see the same gains (+0.16). Translating into English, we saw limited gains, but OSM with classes reduced std. deviation >1.0 BLEU.

Lang.	Baseline	100	250	500	1000
en-fr	41.39	41.56	40.21	–	–
zh-en	12.85	12.89	12.76	12.83	12.98

Table 6: nClasses with OSM w/WCs, `tst2012` for en-fr, `tst2013` for zh-en.

2.7. Russian Morphological Preprocessing

We used a variation of the Yandex technique for reducing data sparsity [21], stemming nouns and adjectives and inserting a case element as a separate word before each noun. We used `mystem`[5] to identify lemmas and grammatical information; nouns were annotated for number, and adjectives were annotated for degree. Noun forms that could represent singular or plural were annotated as singular. For nouns with ambiguous case, the first possible case element was selected from the continuum of nominative, accusative, genitive, dative, instrumental, ablative. Examples are shown in Table 7.

Noun	Case/Number	Output
дням	dat-pl	DAT день.N+PL
день	nom-sg, acc-sg	NOM день.N+SG

Table 7: Examples of Yandex-style morphological processing

Table 8 shows average BLEU gains over 10 runs by preprocessing the Russian source data in this way. Max scores increased less, on average 0.27, while standard deviation decreased significantly. These trends extended to experiments with extra data, and were exaggerated with the addition of NNJM rescoring.

[5] https://api.yandex.ru/mystem/

System	BLEU	Gain	Δ Stdev
Baseline	21.13	+0.32	-0.2
+outd	23.29	+0.82	-0.3
+RescoreNNJM	23.56	+1.45	-1.14

Table 8: Mean BLEU scores with Yandex-style preprocessing, `tst2013`.

2.8. Farsi-English System

Our system this year was a factored phrase-based system built using supplied in-domain data for the phrase table with 3 language models built using Gigaword, in-domain data, and Google-book n-grams. Gains were obtained by replacing non-printable characters with spaces, utilizing class-factors with 600 classes, using the cleaned test sets as described in Section 2.2, and optimizing with a development set as described in Section 2.9. We selected the number of sentences for these sets based on the maximum Tversky score. Three sets were created, one each to match `tst2013` and `tst2014` and one to match the combination. Non-printing characters were replaced and repeated phrases (Section 2.2) removed before the devset selection occurred. Systems were optimized with PRO using each of these devsets and the best score on `tst2012` of 10 runs was selected as the configuration for submission (see Table 9).

Dev Set	Length	`tst2012`		
		Mean	Stdev	Max
`dev2010`	885	20.52	0.22	20.16
`tst2012`	1375	20.48	0.09	20.60
`tst2013devsel`	931	20.94	0.16	21.23
`tst2014devsel`	888	21.22	0.10	21.34
`tst2014+13devsel`	1245	20.99	0.16	21.23

Table 9: Farsi-English system BLEU scores on regular and Tversky-selected devsets

Based on these results, the system optimized with `tst2014devsel` was used to decode `tst2013` and `tst2014` for submission.

2.9. Development Set Creation

Following the experiments from last year, as well as uncertainty in performance via optimizing `dev2010` or `tst2011`, we implemented a dev set creation mechanism which extracts the most promising segments from the available data. We choose to select the dev set based on maximizing the Tversky similarity measure [26] between the dev set source segments and the test set source segments. We employ Tversky similarity with unit weights, making it equivalent to Jaccard similarity and Tanimoto similarity: our Tversky score is the number of unique words in the intersection of the dev and test sets divided by the number of unique words in the union.

We create the dev set via greedy optimization. Starting with an empty dev set, we iteratively add the segment that provides the largest bang-for-your-buck improvement, i.e., the largest increase in Tversky similarity divided by the number of words in the segment. The result is a dev set with segments ordered by relationship to the test set. We can choose a fixed dev set size based on available resources, a dev set size that maximizes Tversky similarity, or use another heuristic.

In order to test effectiveness of the Tversky metric, baseline systems were trained using only in-domain data for Arabic-English, Russian-English, and Chinese-English language pairs. These systems were then optimized using `dev2010`, `tst2012` and Tversky-selected dev sets of varying length (e.g. `tvdev1188` for Arabic indicating a dev set selected from the first 1,188 lines of the selected data). The pool of possible sentence pairs for the Tversky-selected dev sets is the concatenation of `dev2010`, `tst2010`, `tst2011`, and `tst2012`. The length of these selected sets is set by maximizing the score for the source-side of `tst2014`. (It is worth mentioning that the references play no role in the entire process.) Results are shown in Table 10.

Lang.	dev set	avg BLEU	max BLEU
ar-en	`dev2010`	20.42	20.96
	`tst2012`	20.64	20.94
	`tvdev1188`	21.15	21.52
ru-en	`dev2010`	16.98	17.03
	`tst2012`	16.81	17.03
	`tvdev2500`	17.00	17.13
zh-en	`dev2010`	12.60	12.90
	`tst2012`	12.33	13.03
	`tvdev1500`	11.30	12.92

Table 10: Results of Baseline systems using standard and Tversky-score selected dev sets.

2.10. MT Submission Systems

A brief description and results for all of our MT submission systems can be found in Table 11.

3. ASR

3.1. English ASR

A hybrid Deep Neural Network (DNN)-HMM speech recognition system was developed on 166 hours of TED data, 128 hours from the HUB4 corpus [27, 28], and 96 hours from the Euronews corpus provided by the organizers. This system was trained using the same procedure as our IWSLT 2013 system [6]. The DNNs included 7 hidden layers with 1000 units each and 8000 output units. Compared to our IWSLT 2013 hybrid

System	Description	tst2012	tst2013	tst2014
English-to-French				
primary	cE apw/afp/ted/news LMs, NNJMout+in, OSM o9, opt tvDev1500	42.62		
contrast1	primary – tvDev + opt dev2010	41.80		
Arabic-to-English				
primary	2PTs, hier-msd, nyt+news LM, NNJMin, ted-200 cLM, nyt-600 cLM	30.86	31.80	27.70
contrast1	primary – dev2010 + opt tvDev1200	31.11	31.72	27.39
Chinese-to-English				
primary	nyt LM, dLimit-8, hier-msd LR, max sent len 32	13.83	15.67	12.90
contrast1	primary + ltw LM + 150 classes GIZA	14.20	15.44	13.25
contrast2	primary + tvDev1500	14.09	15.43	12.92
contrast3	primary + hier-mslr LR	13.28	15.59	12.64
Farsi-to-English				
primary	PRO, cleaned source data, 600 cLM o7, hiero LR reordering, nyt LM o7, google book o5, opt tvdev2014	21.13	19.49	18.45
contrast1	primary – tvdev2014 + opt tvdev2013	21.12	19.24	18.56
contrast2	primary – tvdev2014 + opt tvdev2013+2014	21.11	19.14	18.27
Russian-to-English				
primary	PRO, cE PT, ted LM o7, outd LM o7, giga LM o5, ted+outd cLM o7, giga nyt cLM o7, yandex parsing, NNJMout+in, opt on dev2010	21.30	24.42	19.45
contrast1	primary – yandex parsing	21.27	24.10	19.08

Table 11: MT Submission Systems.

DNN-HMM system trained on TED, the additional data yielded a 1.2% Word Error Rate (WER) improvement on dev2012 prior to LM rescoring, and a 0.4% WER improvement after LM rescoring.

A bottleneck [29] DNN system for use with a tandem GMM-HMM [30] was trained using 135 hours of TED data. The Theano library for Python [25] was leveraged during DNN training to enable use of the GPU. The final DNN had 4 hidden layers with 1000 units, plus an additional bottleneck layer with 60 units placed between the last two hidden layers. The DNN was trained with 12 Perceptual Linear Prediction features, along with the zeroth coefficient and first, second, and third order differentials. Features were combined with a frame window of 13 to give a total input size of 676. Outputs corresponded to 6000 shared states. A minibatch size of 256 and initial learning rate of 0.3 was used for training the DNN. The "newbob" learning rate schedule as used in [31] was followed.

A tandem GMM-HMM was trained with the bottleneck features, which were run through PCA. The final tandem model included approximately 7000 shared states with 32 Gaussians per state. This sytem did not perform as well as the hybrid system, but was successful in system combination.

LM data selection was implemented using the same procedure as our IWSLT 2012 system [32]. Interpolated trigram and 4-gram LMs were estimated on TED, 1/8 of Gigaword, and 1/8 of News 2007–2013 using the SRILM Toolkit [20].A Recurrent Neural Network (RNN) maximum entropy LM was estimated on the same set of training texts using the RNNLM Toolkit [4]. The network included 160 hidden units, 300 classes in the output layer, 4-gram features for the direct connections, and a hash size of 10^9. The LM vocabulary included 100000 words.

In addition to the hybrid DNN-HMM and tandem systems described above, we also used our IWSLT 2013 HMM acoustic models (AMs) with the updated LMs. This system was cross adaptated using the initial transcripts from the hybrid DNN-HMM system.

Automatic segmentation of the test data was performed using the same procedure as IWSLT 2013. Recognition lattices were produced for each system and then rescored with the interpolated 4-gram LM. Next, 1000-best lists were extracted from each lattice and rescored with the RNN LM. The final LM scores were obtained by linearly interpolating the log probabilities from the 4-gram and RNN LM. Lastly, system combination was perfomed using N-best ROVER.

Table 12 shows the WER of each system on dev2012 after evaluating the second pass decoder, rescoring with the 4-gram LM, and interpolating the 4-gram and RNN LM scores. Note that the first pass hybrid DNN-HMM and tandem systems yielded a 16.7% and 23.1% WER on dev2012, respectively. N-best ROVER of all three systems yielded a 12.4% WER.

3.2. Italian ASR

An Italian pronunciation dictionary was manually created for the most frequent 28000 words from the Eu-

System	Decode-2	4-gram	4-gram + RNN
DNN-HMM	14.8	14.2	13.3
HMM-2013-AM	15.3	14.6	13.7
Tandem	20.8	20.0	18.3

Table 12: English `dev2012` WER. Results are shown for each system after evaluating the second pass decoder, rescoring with the 4-gram LM, and interpolating the 4-gram and RNN LM scores.

ronews corpus. This was done by a member of our group who speaks Italian as a second language. The 51 phone set included 24 non-geminated consonants, 20 geminated consonants, and 7 vowels. A second pronunciation dictionary with 32 phones was created by ignoring gemination.[6] Lastly, a multilingual (ML) pronunciation dictionary was created from the Italian dictionary that ignored gemination and version 0.7a of the English CMU pronunciation dictionary. Italian and English phones were merged when they shared the same IPA symbol;[7] this dictionary included 48 phones.

HMM and hybrid DNN-HMM systems were trained on the Euronews Italian data set using the same procedure as the English systems. One HMM system was trained using the 51 phone set (denoted as HMM-51), and a second HMM system was trained using the the 32 phone set (denoted as HMM-32). HMM-51 included 6000 shared states with an average of 28 mixtures per state, and HMM-32 included 4000 shared states with an average of 24 mixtures per state. The hybrid DNN-HMM system was developed using HMM-51, and the DNNs included 3 hidden layers with 1000 units each and 6000 output units. A final HMM system (denoted as HMM-ML) was developed on Euronews Italian and TED English using the ML pronunciation dictionary; HMM-ML included 6000 shared states with an average of 28 mixtures per state.

Interpolated trigram and 4-gram LMs were estimated on the provided TED training data, Google Books Ngram corpus, and Web 1T 5-gram corpus. Words from the TED data set were split on apostrophes, and N-grams from Google Books were ignored if the source was published prior to the year 2000. The LM vocabulary included 100000 words. An RNN maximum entropy LM was estimated on TED using the RNNLM Toolkit. The network included 320 hidden units, 300 classes in the output layer, 4-gram features for the direct connections, and a hash size of 10^9.

Initial segments of the test data were created using the English neural network SAD. On the `dev2014` partition, it was discovered that the SAD was misclassifying non-speech sections as speech on several TEDx talks. To alleviate this problem, we reprocessed any speech segment longer than 20 seconds with a second SAD that was trained on English telephone speech from the Fisher corpus [33].

Each system was evaluated using HDecode and LM rescoring was performed using the same procedure described in Section 3.1. Cross adaptation was applied to the HMM systems using the initial transcripts from the hybrid DNN-HMM system. The final hypothesis was selected via N-best ROVER of the DNN-HMM, HMM-32 and HMM-ML systems. This combination yielded a 29.5% WER on `dev2014`. Table 13 shows the WER at each decoding stage; for comparison purposes, we have included the results obtained without cross adaptation of the HMM systems.

3.3. ASR Submission Systems

Final submissions on English `tst2014` and `tst2013` and Italian `tst2014` are shown in Table 14.

4. Acknowledgements

The authors would like to thank Tina May and Wahid Abdul Qudus for their efforts in spot-checking Chinese and Farsi dataset processing, respectively. We would also like to thank Kyle Wilkinson for creating the Italian pronunciation dictionary.

5. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2014 Evaluation Campaign," ser. Proceedings of IWSLT, 2014.

[2] A. Birch, N. Durrani, and P. Koehn, "Edinburgh SLT and MT system description for the iwslt 2013 evaluation," *Proc. IWSLT, Heidelberg, Germany*, 2013.

[3] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," ser. Proceedings of the ACL, Long Papers, 2014.

[4] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocký, "Strategies for training large scale neural network language models," ser. Automatic Speech Recognition and Understanding Workshop, 2011.

[6] Palatal nasal consonants were always geminated in our dictionary.

[7] The ARPAbet to IPA mappings used in this work are available at: http://en.wikipedia.org/wiki/Arpabet

Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government. Cleared for public release on 6 Nov 2014. Originator Reference Number: SA-14-113170. Case Number: 88ABW-2014-5156.

	Without Cross Adaptation					With Cross Adaptation		
System	Decode-1	Decode-2	4-gram	4-gram + RNN		Decode-2	4-gram	4-gram + RNN
DNN-HMM	35.0	32.9	32.5	32.5		32.9	32.5	32.5
HMM-32	41.2	34.4	34.1	33.9		32.2	31.8	31.4
HMM-51	41.2	35.1	34.8	34.5		32.2	31.9	31.8
HMM-ML	42.7	35.9	35.7	35.4		32.4	32.3	32.3
N-best ROVER	35.2	31.3	30.8	30.8		30.1	29.7	29.5

Table 13: Italian dev2014 WER. N-best ROVER was applied at each decoding stage using 1000-best lists from the the hybrid DNN-HMM, HMM-32, and HMM-ML systems. Results are shown both with and without cross adaptation of the HMM systems.

System	Description	tst2013	tst2014
English			
primary	N-best ROVER with hybrid DNN-HMM, HMM Sphinx-4, and tandem systems	14.3*	10.0**
contrast1	Hybrid DNN-HMM using Viterbi decoding	15.6*	11.2**
Italian			
primary	N-best ROVER with hybrid DNN-HMM, HMM-32, and HMM-ML systems	–	24.7

Table 14: All submission systems for English and Italian ASR. *Unofficial scores using last year's tst2013 reference files with minor corrections. **Unofficial scores using suggested tst2014 STM and GLM corrections.

[5] N. Durrani, H. Schmid, and A. Fraser, "A joint sequence translation model with integrated reordering," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 1045–1054.

[6] M. Kazi, M. Coury, E. Salesky, J. Ray, W. Shen, T. Gleason, T. Anderson, G. Erdmann, L. Schwartz, B. Ore, R. Slyh, J. Gwinnup, K. Young, and M. Hutt, "The MIT-LL/AFRL IWSLT-2013 MT system," in *The 10th International Workshop on Spoken Language Translation (IWSLT '13)*, Heidelberg, Germany, December 2013, pp. 136–143.

[7] M. Cettolo, C. Girardi, and M. Federico, "WIT3: Web Inventory of Transcribed and Translated Talks," ser. Proceedings of EAMT, 2012, pp. 261–268.

[8] X. Ma, "Champollion: A robust parallel text sentence aligner," ser. Proceedings of LREC, 2006.

[9] R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda, "English Gigaword Fifth Edition LDC2011T07," *Philadelphia: Linguistic Data Consortium*, 2011.

[10] D. Graff, Ângelo Mendonça, and D. DiPersio, "French Gigaword Third Edition LDC2011T10." *Philadelphia: Linguistic Data Consortium*, 2011.

[11] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 48–54.

[12] N. Habash, R. Roth, O. Rambow, R. Eskander, and N. Tomeh, "Morphological analysis and disambiguation for dialectal Arabic." in *HLT-NAACL*. The Association for Computational Linguistics, 2013, pp. 426–432.

[13] P.-C. Chang, M. Galley, and C. D. Manning, "Optimizing Chinese word segmentation for machine translation performance," in *Proceedings of the Third Workshop on Statistical Machine Translation*, ser. StatMT '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 224–232.

[14] G. Foster, R. Kuhn, and H. Johnson, "Phrasetable smoothing for statistical machine translation," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 53–61.

[15] C. Tillmann, "A unigram orientation model for statistical machine translation," in *Proceedings of HLT-NAACL 2004: Short Papers*, ser. HLT-NAACL-Short '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004, pp. 101–104.

[16] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 848–856.

[17] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 177–180.

[18] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 1352–1362.

[19] B.-J. P. Hsu and J. Glass, "Iterative language model estimation: Efficient data structure and algorithms," ser. Interspeech, 2008.

[20] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Proceedings International Conference on Spoken Language Processing*, November 2002, pp. 257–286.

[21] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable modified Kneser-Ney language model estimation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August 2013, pp. 690–696. [Online]. Available: http://kheafield.com/professional/edinburgh/estimate_paper.pdf

[22] K. Heafield, "KenLM: faster and smaller language model queries," in *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom, July 2011, pp. 187–197.

[23] F. J. Och, "An efficient method for determining bilingual word classes," in *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, ser. EACL '99. Stroudsburg, PA, USA: Association for Computational Linguistics, 1999, pp. 71–76.

[24] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the ACL 2010 Conference Short Papers*, ser. ACLShort '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 220–224.

[25] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

[26] A. Tversky, "Features of similarity," *Psychological Review*, vol. 84, pp. 327–352, 1977.

[27] D. Graff, Z. Wu, R. MacIntyre, and M. Liberman, "The 1996 broadcast news speech and language-model corpus," in *Proceedings of the DARPA Workshop on Spoken Language technology*. Citeseer, 1997, pp. 11–14.

[28] J. Fiscus, J. Garofolo, M. Przybocki, W. Fisher, and D. Pallett, "English broadcast news speech (hub4)," *Linguistic Data Consortium, Philadelphia*, 1997.

[29] F. Grezl, M. Karafiát, S. Kontár, and J. Cernockỳ, "Probabilistic and bottle-neck features for LVCSR of meetings." in *ICASSP (4)*, 2007, pp. 757–760.

[30] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 3. IEEE, 2000, pp. 1635–1638.

[31] D. Johnson *et al.*, "ICSI quicknet software package," *http://www.icsi.berkeley.edu/Speech/qn.html*, 2004.

[32] J. Drexler, W. Shen, T. Gleason, T. Anderson, R. Slyh, B. Ore, and E. Hansen, "The MIT-LL/AFRL IWSLT-2012 MT system," ser. Proceedings of IWSLT, 2012.

[33] C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker, "Fisher English training parts 1 and 2, speech and transcripts," *Linguistic Data Consortium, Philadelphia*, 2005.

The 2014 KIT IWSLT Speech-to-Text Systems for English, German and Italian

Kevin Kilgour, Michael Heck, Markus Müller,
Matthias Sperber, Sebastian Stüker and Alex Waibel

Institute for Anthropomatics
Karlsruhe Institute of Technology
Karlsruhe, Germany

{kevin.kilgour|heck|m.mueller}@kit.edu
{matthias.sperber|sebastian.stueker|waibel}@kit.edu

Abstract

This paper describes our German, Italian and English *Speech-to-Text* (STT) systems for the 2014 IWSLT TED ASR track. Our setup uses ROVER and confusion network combination from various subsystems to achieve a good overall performance. The individual subsystems are built by using different front-ends, (e.g., MVDR-MFCC or lMel), acoustic models (GMM or modular DNN) and phone sets and by training on various subsets of the training data. Decoding is performed in two stages, where the GMM systems are adapted in an unsupervised manner on the combination of the first stage outputs using VTLN, MLLR, and cMLLR.

The combination setup produces a final hypothesis that has a significantly lower WER than any of the individual subsystems.

1. Introduction

The 2014 *International Workshop on Spoken Language Translation* (IWSLT) offers a comprehensive evaluation campaign on spoken language translation. The evaluation is organized in different evaluation tracks covering automatic speech recognition (ASR), machine translation (MT), and the full-fledged combination of the two of them into speech translation systems (SLT). The evaluations in the tracks are conducted on TED Talks (http://www.ted.com/talks), short 5-25min presentations by people from various fields related in some way to Technology, Entertainment, and Design (TED) [1].

The goal of the TED ASR track is the automatic transcription of fully unsegmented TED lectures. The quality of the resulting transcriptions are measured in word error rate (WER).

In this paper we describe our Italian, German and English ASR systems with which we participated in the TED ASR track of the 2014 IWSLT evaluation campaign. While our German and English ASR systems are based on our previous years' evaluation systems [2] our Italian system is a completely new system that was developed from scratch. Our general system setup uses multiple complementary subsystems that employ different phone sets, front ends, acoustic models or data subsets.

The rest of this paper is structured as follows. Section 2 describes the data that our system was trained and tested on. This is followed by Section 3 which provides a description of the acoustic front-ends used in our system and Section 4 which describes our segmentation setup. An overview of the techniques used to build our acoustic models is given in section 5. We describe the language model used for this evaluation in section 6. Our decoding strategy and results are then presented in sections 7 and 8. The final section, Section 8 contains a short conclusion.

2. Data Resources

2.1. Training Data

The following data sources have been used for acoustic model training of all our English systems:

- 200 hours of Quaero training data from 2010 to 2012.
- 18 hours of various noise data, such as snippets of applause, music or noises from microphone movement.
- 158 hours of data downloaded from the TED talks website that was released before the cut-off date of December 31st 2010.

The Quaero training data is manually transcribed. The noise data consists only of noises and is tagged with specific noise words to enable the training of noise models. The TED data comes with subtitles provided by TED and the TED translation project.

For German we used the following data sources:

- 180 hours of Quaero training data from 2009 to 2012.
- 24 hours of broadcast news data
- 160 audio from the archive of parliament of the state of Baden-Württemberg, Germany

Set	#talks	#utt	dur	dur/utt
dev2010	8	887	1.5h	6.2s
dev2012	10	1144 (545)	1.7h (1.8h)	5.4s (12.2s)
tst2010	11	1664	2.5h	5.3s
tst2013	28	1388	4.2h	10.8s
tst2014	15	718	2.2h	11.0s

Table 1: *Statistics of the development sets ("dev2010", "tst2010" and "dev2012") and the evaluation sets ("tst2013" and "tst2014"), including the total number of talks (*#talks*), the total number of utterances (*#utt*), the overall speech duration (*dur*), and average speech duration per utterance (*dur/utt*). "tst2013" and "tst2014" have been segmented automatically. Properties of the automatic segmentation of "dev2012" is described in brackets.*

The training database for our Italian system contains a total of 100 hours of audio. It is based on the data from Quaero Period 4 (54 hours) and Quaero Period 5 (46 hours). The audio consists of recordings from radio and TV broadcasts. The data is manually transcribed and split into segments of varying length, ranging from one sentence to multiple minutes. The textual transcriptions contain annotations for distinct acoustic events as well. We incorporated them as markers for noises in general and for noises originating from humans.

Due to the lack of Italian data, we used additional English data for the neural network training. This data consisted of 426 hours, based on a selection of TED talks, stanford lectures, euronews broadcasts and recordings from videolectures.

For language modeling and vocabulary selection, we used most of the data admissible for the evaluation, as summarized in Tables 2, 3, and 4.

2.2. Test Data

For this year's evaluation campaign, two evaluation test sets ("tst2013" and "tst2014") were provided, as well as three development test sets ("dev2010", "tst2010" and "dev2012"). The test set "dev2012" has preferably been used for system development and parameter optimization. Table 1 lists these five test sets along with relevant properties.

"tst2013" is last year's evaluation set and is solely comprised of TED talks newer than December 2010. This set serves as a progress test set to measure the system improvements with respect to last year's IWSLT ASR track. "tst2014" is a collection of TED talks that have been filmed between early 2012 and late 2013. All development test sets were used with the original pre-segmentation provided by the IWSLT organizers. Additionally, "dev2012" has been segmented automatically, as well this year's evaluation test set.

For the German and Italian systems only a single test each set "dev2013" and "dev2014" was available.

3. Feature Extraction

Our systems are built using several different front ends. The two main input variants, each using a frame shift of 10ms and a frame size of 32ms, are the mel frequency ceptral coefficient (MFCC) minimum variance distortionless response (MVDR) (M2) features that have been shown to be very effective when used in BNFs [3] and standard lMEL features which generally outperform MFCCs when used as inputs to deep bottleneck features. These standard features are often augmented by tonal features (T). In [4] we demonstrate, that the addition of tonal features not only greatly reduces the WER on tonal languages like Vietnamese and Cantonese but also results in small gains on non-tonal languages such as English.

For bootstrapping our systems we employed log Mel features with 13 coefficients and a frame size of 16ms. We stacked the individual frames using a context of seven frames to each side.

3.1. Deep Bottleneck Features

The use of bottleneck features greatly improves the performance of our GMM acoustic models. Figure 1 shows a general overview of our deep bottleneck features training setup. 13 frames (+-6 frames) are stacked as the DBNF input which consists of 4-5 hidden layers each containing 1200-1600 units followed by a 42 unit bottleneck, a further 1200-1600 unit hidden layer and an output layer of 6000 context dependent phone states for the German systems and 8000 for the English systems. Layer-wise pretraining with denoising autoencoders is used for the all the hidden layers prior to the bottleneck layer. The network is subsequently finetuned as a whole [5].

The layers following the bottleneck are discarded after training and the resulting network can then be used to map a stream of input features to a stream of 42 dimensional bottleneck features. Our experiments show it to be helpful to stack a context of 13 (+-6) bottleneck features and perform LDA on this 630 dimensional stack to reduce its dimension back to 42.

For Italian, we used an additional approach by training a neural network using data from more than one language. We re-used a neural network that has been trained using English data. In one setting, we used it directly without any re-training and in another setting, we re-added the discarded output layers after the bottleneck and re-trained them using Italian data.

4. Automatic Segmentation

As was the case for last year's evaluation, the test set for the ASR track was provided without manual sentence segmentation, thus automatic segmentation of the target data was mandatory. We utilized three different approaches to automatic segmentation of audio data, which are:

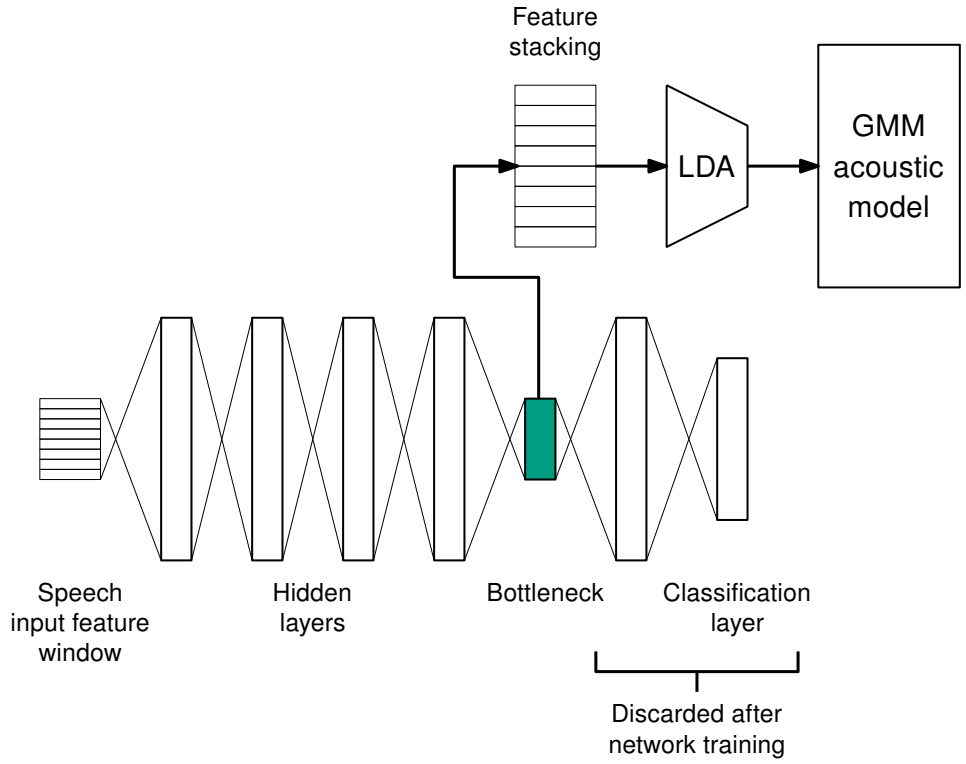


Figure 1: *Overview of our standard DBNF setup.*

a) *Decoder based* segmentation on hypotheses. A fast decoding pass with one of our development systems was done to determine speech and non-speech regions as in [6]. Segmentation is then performed by consecutively splitting segments at the longest non-speech region with a minimal duration of at least 0.3 seconds. *b*) *GMM based* segmentation using speech, non-speech and silence models. This method uses a Viterbi decoder and MFCC GMM models for the three aforementioned categories of sounds. The general framework is based on the one in [7], which was likewise derived from [8]. In contrast to the previous work, we made use of additional features such as a zero crossing rate. *c*) *SVM based* segmentation using speech and non-speech models, using the framework introduced in [7]. The pre-processing makes use of an LDA transformation on DBNF feature vectors after frame stacking to effectively incorporate temporal information. The SVM classifier is trained with the help of LIBSVM [9]. A 2-phased post-processing is applied for final segment generation.

We generated the segmentation of the English data with the decoder based approach. Our German data was segmented with the help of the SVM based segmentation. The data for the Italian track was pre-processed using the GMM framework. The decisions for the respective segmenters have been made in accordance to previous experiments and successful usages within the frame of various projects.

5. Acoustic Modeling

5.1. Data Preprocessing

For the TED data only subtitles were available so the data had to be segmented prior to training. In order to split the data into sentence-like chunks, it was decoded by one of our development systems to discriminate speech and non-speech and a forced alignment given the subtitles was performed where only the relevant speech parts detected by the decoding were used. The procedure is the same as the one that has been applied in [10].

5.2. GMM AM training Setup

All systems use context-dependent quinphones with three states per phoneme and a left-to-right HMM topology without skip states. The English and Italian acoustic models use 8000 distributions and codebooks derived from decision-tree based clustering of the states of all possible quinphones. The German acoustic models use 6000 distributions and codebooks.

The GMM models are trained by using incremental splitting of Gaussians training (MAS) [11], followed by optimal feature space training (OFS) which is a variant of *semi-tied covariance* (STC) [12] training using a single global transformation matrix. The model is then refined by one iteration of Viterbi training. All models further use vocal tract length

normalization (VTLN).

In order to improve the performance of our acoustic model Boosted Maximum Mutual Information Estimation training (BMMIE) [13], a modified form of the Maximum Mutual Information (MMI) [14], is applied at the end. Lattices for discriminative training use a small unigram language model as in [15]. After lattice generation, the BMMIE training is applied for three iterations with a boosting factor of b=0.5. This approach results in about 0.6% WER improvement for 1st-pass sytems and about 0.4% WER for 2nd-pass systems.

We trained multiple different GMM acoustic models by combining different front-ends and different phoneme sets. Section 7 elaborates the details of our system combination.

In contrast to our systems for English and German, we did not have an existing system for Italian, hence we bootstrapped our acoustic model using a flatstart training technique to acquire the initial models.

5.3. Hybrid Acoustic Model

As with the GMM systems we trained our hybrid systems on variance front-ends and phoneme sets. Our best performing hybrid systems are based on a modular topology which involves stacking the bottleneck features, described in the previous section over a window of 13 frames, with 4-5 1600-2000 unit hidden layers and an output layer containing 6016 context dependent phonestates. The deep bottleneck features were extracted using an MLP with 5 1600 unit hidden layers prior to the 42 unit bottleneck layer. Its input was 40 lMel (or MVDR+MFCC) and 14 tone features stacked over a 13 frame window. Both neural networks were pretrained as denoising autoencoders.

5.4. Pronunciation Dictionary

For Italian, we used a pronunciation dictionary which is based on SAMPA, including consonant geminates and pronunciation variants. It contains 55 phonemes including noises and consists of the 100k words from the search vocabulary.

For our English systems we used two different phoneme sets. The first one is based on the CMU dictionary[1] and is the same phoneme set as the one used in last year's system. It consists of 45 phonemes and allophones. The second phoneme set is derived from the BEEP dictionary[2] and contains 44 phonemes and allophones. Both sets use 7 noise tags and one silence tag each. For the CMU phoneme set we generated missing pronunciations with the help of FESTIVAL [16], while for the BEEP dictionary we used Sequitur [17] instead. Both grapheme to phoneme converters were trained on subsets of the respective dictionaries.

Our German system uses an initial dictionary based on the Verbmobil Phoneset [18]. Missing pronunciations are generated using both Mary [19] and FESTIVAL [16].

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[2] ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/dictionaries/beep.tar.gz

5.5. Grapheme System

In addition to systems with a phoneme-based dictionary, we also built grapheme-based recognition systems for both German and Italian. By using a different set of phones, grapheme based systems are an additional source of information when doing system combination. Such systems do not require a pronunciation dictionary, as a 1:1 mapping approach between letters and sounds is used. Depending on the language, the resulting system suffers in performance as this naive approach of letter to sound mapping does not reflect any pronunciation rules.

As the pronunciation of Italian is known to be close to a 1:1 mapping, the Italian system performed only slightly worse compared to the phoneme-based system and including it into system combination resulted in overall gains. The German grapheme systems had about a 1% absolute lower WER than an equivalent phoneme system.

6. Language Models and Search Vocabulary

For language model training and vocabulary selection, we used the subtitles of TED talks, or translations thereof, and text data from various sources (see Tables 2, 3, and 4). Language model training was performed by building separate language models for all (sub-)corpora using the SRILM toolkit [20] with modified Kneser-Ney smoothing. These were then linearly interpolated, with interpolation weights tuned using held-out data from the TED corpus. For Italian, we attempted to compensate for the small amount of data by using a more elaborate language model with data selected via Moore's method [21], but observed no significant improvement in terms of word error rate. For German, we split compounds similarly as in [22].

For the vocabulary selection, we followed an approach proposed by Venkataraman et al.[23]. We built unigram language models using Witten-Bell smoothing from all text sources, and determined unigram probabilities that maximized the likelihood of a held-out TED data set. As our vocabulary, we then used the top 150k words for English, 300k words for German, and 100k words for Italian.

7. Decoding Setup

For the evaluation, we built four final systems for Italian. Three are based on the phoneme dictionary. One is using a neural network trained entirely on English for feature extraction, one is using a neural network that was pre-trained on English but fine-tuned on Italian and the last one is using a feature front-end with just lMEL features. A fourth system is based on a grapheme dictionary and uses a network that was trained entirely on English.

Our primary submission is a confusion network combination (CNC) using all three phoneme-based systems. The first contrastive system uses the phoneme dictionary and the

Text corpus	# Words
TED	3m
News + News-commentary + -crawl	4,478m
Euronews	780k
Commoncrawl	185m
GIGA	2323m
Europarl + UN + multi-UN	829m
Google Books	(1b n-grams)

Table 2: *English language modeling data after cleaning. The total number of words was 7.8 billion, not counting Google Books.*

Text corpus	# Words
TED	2,685k
News+Newscrawl	1,500M
Euro Language Newspaper	95,783k
Common Crawl	51,156k
Europarl	49,008k
ECI	14,582k
MultiUN	6,964k
German Political Speeches	5,695k
Callhome	159k
HUB5	20k
Google Web	(118m n-grams)

Table 3: *German language modeling data after cleaning and compound splitting. In total, we used 1.7 billion words, not counting Google Ngrams.*

network that was trained using only English data. The second contrastive system is based on graphemes and is using the same neural network. Our third contrastive system is a ROVER of the two phoneme-based systems using a neural network and the grapheme-based system using the network trained on English entirely.

For our English submission we trained 5 different DBNF GMM acoustic models in total by combining different feature front-ends (M2 and lMEL) and different phoneme sets (CMU and BEEP). In addition to these systems, we trained 2 DBNF DNN hybrid systems, one for each phoneme set. For our primary submission, we combined all 7 systems in a

Text corpus	# Words
TED	3,050k
ECI	480k
Euronews	725k
Google Books	(437m n-grams)

Table 4: *Italian language modeling data after cleaning and data selection. The total number of words was 4.3 million, not counting Google Books.*

System	Dev
lMel+FFV+Pitch EN-NN	38.4
lMel+FFV+Pitch EN-NN Grapheme	38.7
lMel+FFV+Pitch EN-NN IT-ft	40.7
lMel	40.8
ROVER	37.4
CNC	37.1

Table 5: *Italian language results on development data (dev2014)*

CNC. The 5 DBNF GMM systems were adapted in an unsupervised manner on the combination of the first stage outputs using VTLN, MLLR, and cMLLR. A second CNC was computed using the adapted systems and the 2 unadapted hybrid systems. The final submission consists of a ROVER of both CNCs, the 5 adapted DBNF GMM systems and the 2 hybrid systems.

The German setup consisted of 9 separate subsystems 5 with discriminativly trained GMM acoustic models (**bmmie**) and 4 using DNN acoustic models (**hyb**). A confusion network combination is performed on the output of these 9 systems which is then used to adapt the 5 GMM based acoustic models for which a 2nd pass speaker adaped pass is then performed. In the 2nd confusion network combination the 2nd pass systems replace the orginal GMM systems. A ROVER of the hybrid systems, the 2nd pass GMM system and both CNCs results in the final output.

8. Results

Our German evaluation setup has improved noticeably since last year from 18.3% to 17.6% (see Table 7). The best first pass system now has a WER of 19.2%, an improvement of 0.8% abs. over last year's best first pass system. The best 2nd pass system has improved by 1.0% abs.

We evaluated our Italian system on the 2014 dev set (dev2014). Tabel 5 shows the results for different single systems and ROVER and CNC combinations.

The English system has been evaluated on the test sets "dev2012". The results are listed in Table6.

9. Conclusions

In this paper we presented our Italian, English and German LVCSR systems, with which we participated in the 2014 IWSLT evaluation. All systems make use of neural network based front-ends, HMM/GMM and HMM/DNN based acoustics models. The decoding set-up of all languages makes extensive use of system combination of single systems obtained by combing different phoneme sets, feature extraction front-ends and acoustic models.

In German we were able to considerably improve the system over last year's system. For Italian we created for the first time a large scale Italian speech recognition system for

System	dev2012
M2+T-CMU	15.7
lMEL+T-CMU	15.5
M2+T-16ms-CMU	15.9
M2+T-BEEP	16.0
lMEL+T-BEEP	16.2
lMEL+T-hyb-CMU	15.9
lMEL+T-hyb-BEEP	16.7
CNC-BEEP-01	13.4
M2+T-CMU	14.3
lMEL+T-CMU	14.4
M2+T-16ms-CMU	14.8
M2+T-BEEP	14.6
lMEL+T-BEEP	14.5
CNC-BEEP-02	13.5
ROVER	13.4

Table 6: *Results for English on development test sets.*

evaluation purposes.

10. Acknowledgements

The authors which to thank Roberto Gretter for providing an Italian pronunciation dictionary for us. The work leading to these results has received funding from the European Union under grant agreement $n \circ 287658$.

11. References

[1] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 10th iwslt evaluation campaign," in *Procedings of the 10th Workshop on Spoken Language Translation (IWSLT 2013)*, 2013.

[2] Christian Saam, Christian Mohr, Kevin Kilgour, Michael Heck, Matthias Sperber, Keigo Kubo, Sebastian Stüker, Sakriani Sakti, Graham Neubig, Tomoki Toda, Satoshi Nakamura, and lex Waibel, "The 2012 KIT and KIT-NAIST English ASR Systems for the IWSLT Evaluation," in *International Workshop on Spoken Language Translation (IWSLT)*, Dec. 2012.

[3] K. Kilgour, I. Tseyzer, Q. B. Nguyen, and A. Waibel, "Warped minimum variance distortionless response based bottle neck features for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6990–6994.

[4] F. Metze, Z. A. W. Sheikh, A. Waibel, J. Gehring, K. Kilgour, Q. B. Nguyen, and V. H. Nguyen, "Models of tone for tonal and non-tonal languages," in *Procedings of the 10th Workshop on Spoken Language Translation (IWSLT 2013)*, 2013.

System	Dev2012
lMEL-all-hyb-P	19.4
lMEL-nl-hyb-P	19.2
M2+T-G-bmmie	21.0
M2-hyb-P	20.4
lMEL+T-P-bmmie	20.2
lMEL-hyb-P	19.3
M2-G-bmmie	22.2
M2-P-bmmie	20.3
M2+T-P-bmmie	20.0
CNC1	17.9
M2+T-G-bmmie	19.5
lMEL+T-P-bmmie	19.0
M2-G-bmmie	20.9
M2+T-P-bmmie	18.7
M2-P-bmmie	19.3
CNC2	17.6
ROVER	17.6
2013 setup	18.3
best 2013 1. pass	20.0
best 2013 2. pass	19.7

Table 7: *Results for German language on development data. Systems designated with **M2** use MFCC+MVDR features, **lMEL** systems use log Mel feature and **+T** means that the system also uses tonal features. Hybrid systems are marked with **hyb** with **bmmie** corresponding to systems using bmmie trained GMM acoustic models. Some systems are phoneme based **P** while others are grapheme based **G**.*

[5] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE*, 2013.

[6] S. Stüker, C. Fügen, F. Kraft, and M. Wölfel, "The ISL 2007 English Speech Transcription System for European Parliament Speeches," in *Proceedings of the 10th European Conference on Speech Communication and Technology (INTERSPEECH 2007)*, Antwerp, Belgium, August 2007, pp. 2609–2612.

[7] M. Heck, C. Mohr, S. Stker, M. Mller, K. Kilgour, J. Gehring, Q. Nguyen, V. Nguyen, and A. Waibel, "Segmentation of telephone speech based on speech and non-speech models," in *Speech and Computer*, ser. Lecture Notes in Computer Science, M. elezn, I. Habernal, and A. Ronzhin, Eds. Springer International Publishing, 2013, vol. 8113, pp. 286–293.

[8] H. Yu, Y.-C. Tam, T. Schaaf, S. Stüker, Q. Jin, M. Noamany, and T. Schultz, "The ISL RT04 Mandarin Broadcast News Evaluation System," in *EARS Rich Transcription Workshop*, 2004.

[9] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[10] M. Heck, K. Kubo, M. Sperber, S. Sakti, S. Stker, C. Saam, K. Kilgour, C. Mohr, G. Neubig, T. Toda, S. Nakamura, and A. Waibel, "The KIT-NAIST (contrastive) english ASR system for IWSLT 2012," in *Proceedings of the International Workshop on Speech Translation (IWSLT 2012)*, Hong Kong, December 2012.

[11] T. Kaukoranta, P. Fränti, and O. Nevalainen, "Iterative split-and-merge algorithm for VQ codebook generation," *Optical Engineering*, vol. 37, no. 10, pp. 2726–2732, 1998.

[12] M. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[13] Daniel Povey, Dimitri Kanevsky, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Karthik Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *ICASSP 2008*, 2008, pp. 4057–4060.

[14] Bahl L.R., Brown P.F, de Souza P.V., and L.R. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *ICASSP 1986*, 1986, pp. 49–52.

[15] V. Valtchev, J. J. Odell, P.C. Woodland, and S.J. Young, "MMIE training of large vocabulary recognition systems," in *Speech Communication 22*, 1997, pp. 303–314.

[16] A. Black, P. Taylor, R. Caley, and R. Clark, "The festival speech synthesis system," 1998.

[17] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Commun.*, vol. 50, no. 5, pp. 434–451, May 2008. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2008.01.002

[18] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, and M. Westphal, "The karlsruhe-verbmobil speech recognition engine," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 83–86.

[19] M. Schröder and J. Trouvain, "The german text-to-speech synthesis system mary: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.

[20] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh International Conference on Spoken Language Processing*, 2002.

[21] R. C. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *Proceedings of ACL*, 2010.

[22] Kevin Kilgour, Christian Mohr, Michael Heck, Quoc Bao Nguyen, Van Huy Nguyen, Evgeniy Shin, Igor Tseyzer, Jonas Gehring, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel, "The 2013 KIT IWSLT Speech-to-Text Systems for German and English," in *International Workshop on Spoken Language Translation (IWSLT)*, Dec. 2013.

[23] A. Venkataraman and W. Wang, "Techniques for effective vocabulary selection," in *Proceedings of the 8th European Conference on Speech Communication and Technology*, 2003, pp. 245–248.

A Topic-based Approach for Post-processing Correction of Automatic Translations

Mohamed Morchid, Stéphane Huet, Richard Dufour

Laboratoire Informatique d'Avignon (LIA)
University of Avignon, France
firstname.lastname@univ-avignon.fr

Abstract

We present the LIA systems for the machine translation evaluation campaign of the *International Workshop on Spoken Language Translation* (IWSLT) 2014 for the English-to-Slovene and English-to-Polish translation tasks. The proposed approach takes into account word context; first, it maps sentences into a latent Dirichlet allocation (LDA) topic space, then it chooses from this space words that are thematically and grammatically close to mistranslated words. This original post-processing approach is compared with a factored translation system built with MOSES. While this post-processing method does not allow us to achieve better results than a state-of-the-art system, this should be an interesting way to explore, for example by adding this topic space information at an early stage in the translation process.

1. Introduction

This paper presents an original post-processing approach to correct machine translations using a set of topic-based features. The proposed method proceeds after the use of factored phrase-based machine translation (MT) systems [1]. The post-processed systems were submitted at the IWSLT 2014 MT evaluation campaign for two language directions: English-to-Slovene and English-to-Polish.

The focus and the major contribution of the proposed approach lie on mapping sentences to a topic space learned from a latent Dirichlet allocation (LDA) model [2], in order to replace every word identified as mistranslated with a thematically and grammatically close word. The idea behind this approach is that during the LDA learning process, the words contained in each sentence will retain the grammatical structure. Indeed, a topic space is usually learned from a corpus of documents and each document is considered as a "bag-of-words". Thus, the structure of sentences is lost as opposed to the proposed topic space that is learned from a corpus of sentences instead. This new topic space takes into account word distribution into sentences and is able to infer classes of close words.

In this exploratory study, the topic-based approach is applied in the context of automatic translations of morphologically rich languages. Slovene and Polish are both Slavic languages which are characterized by many inflections for a great number of words to indicate grammatical differences. This introduces many forms for a same lemma and raises many difficulties when translating from morphologically poor languages such as English. To deal with this problem in this study, words identified as erroneous are replaced by the morphological variant form sharing the same lemma and having the highest LDA score.

We summarize in Section 2 the resources used and the main characteristics of our systems based on the MOSES toolkit [3]. Section 3 presents the proposed topic-based approach to correct mistranslated words. Section 4 reports experiments on the use of factored translation models and the proposed approach. Finally, conclusions and perspectives are given in Section 5.

2. MOSES System Based on Factored Translation Models

2.1. Pre-processing

Systems were only built using data provided for the evaluation campaign, *i.e.* the *WIT* and *Europarl* corpora. Texts were pre-processed using an in-house script that normalizes quotes, dashes and spaces. Long sentences or sentences with many numeric or non-alphanumeric characters were also discarded. Each corpus was truecased, *i.e.* all words kept their case, apart from sentence-leading words that may be changed to their most frequent form (*e. g.* "Write" becomes "write" while "Paris" keeps its capital letter). Table 1 summarizes the used data and introduces designations that we follow in the remainder of this paper to refer to these corpora.

Slovene and Polish are morphologically rich languages with nouns, adjectives and verbs inflected for case, number and gender. This property requires to introduce morphological information inside the MT system to handle the lack of many inflectional forms inside training corpora. For this purpose, each corpus was tagged with Part-of-Speech (PoS) tags and lemmatized using OBELIKS [4] for Slovene[1] and TREETAGGER [5] for Polish[2]. These taggers asso-

[1] OBELIKS can be downloaded at http://eng.slovenscina.eu/tehnologije/oznacevalnik.

[2] TREETAGGER and its parameter file for Polish can be downloaded

CORPORA	DESIGNATION	SIZE (SENTENCES)
English-Slovene bilingual training		
Web Inventory of Transcribed and Translated Talks	*WIT*	17 k
Europarl v7	*Europarl*	616 k
English-Slovene development and test		
dev2012	*dev*	1.1 k
tst2012	*test0*	1.4 k
tst2013	*test13*	1.1 k
tst2014	*test14*	0.9 k
English-Polish bilingual training		
Web Inventory of Transcribed and Translated Talks	*WIT*	173 k
Europarl v7	*Europarl*	622 k
English-Polish development and test		
dev2010	*dev*	0.8 k
tst2010	*test0*	1.6 k
tst2013	*test13*	1.0 k
tst2014	*test14*	1.2 k

Table 1: Information on corpora.

ciate each word with a complex PoS including morphological information (e.g. "Ncmsan" for "Noun Type=common Gender=masculine Number=singular Case=accusative Animate=no"), and also its lemma. A description of the Slovene and Polish tagsets can be found on the Web[3].

In order to simplify the use of the two PoS taggers, we applied the tokenizer included in the OBELIKS and TREETAGGER tools to process all the corpora.

2.2. Language Models

Kneser-Ney discounted LMs were built from the Slovene and Polish sides of the bilingual corpora using the SRILM toolkit [6]. 4-gram LMs were trained for words, 7-gram LMs for PoS. A LM was built separately on each corpus: *WIT* and *Europarl*. These LMs were combined through linear interpolation. Weights were fixed by optimizing the perplexity on the *dev* corpus.

2.3. Alignment and Translation Models

All parallel corpora were aligned using MGIZA++ [7]. Our translation models are phrase-based models (PBMs) built with MOSES using default settings on a bilingual corpus made of *WIT* and *Europarl*. Weights of LM, phrase table and lexicalized reordering model scores were optimized on *dev* with the MERT algorithm [8].

at `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger`.

[3]See `http://nl.ijs.si/spook/msd/html-en/msd-sl.html` for Slovene and `http://nkjp.pl/poliqarp/help/ense2.html` for Polish.

2.4. Factored Translation Model

The many inflections for Slovene and Polish are problematic for translation since morphological information, including case, gender and number, has to be induced from the English words. Factored translation models can be used to handle morphology and PoS during translations [1], with various setups available to use factors in several decoding or generation steps. In previous experiments conducted on translation into Russian, another morphologically rich language [9], we found that translating English words into (Russian words, PoS) pairs gave the highest improvements. We decided to apply this setup, which disambiguates translated words according to their PoS, for Slovene and Polish.

3. Post-processing Approach Relying on LDA

Classical language models consider words in their context (*n-gram*). Nonetheless, all possible contexts cannot be covered and some n-grams contained in the test corpus may not appear during the training process of the language model. For this reason, we propose to learn a topic space using LDA to associate a word inside a sentence with a set of thematically close words. By thematically, we mean that this word is associated with the context of the words contained in the sentence. Indeed, when a topic space is learned from a corpus of documents with usual LDA, words are associated with a document while grammatical structure is lost. In our case, this structure is preserved. Figure 1 gives an overview of the proposed topic-based approach to correct mistranslated words.

The next sections describe each step of the proposed approach based on a LDA topic space.

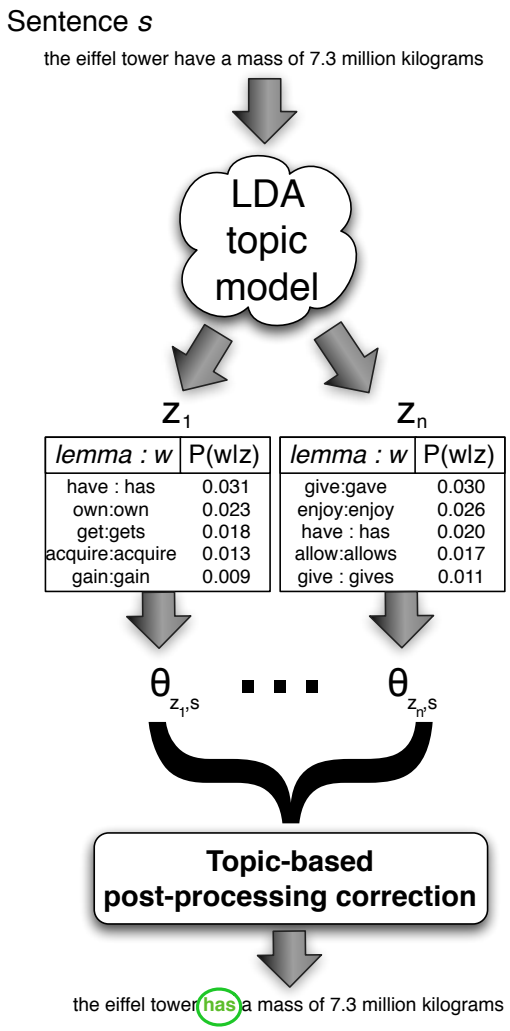


Figure 1: General overview of the proposed post-processing topic-based correction approach.

3.1. Latent Dirichlet Allocation (LDA)

Previous studies proposed to consider a document as a mixture of latent topics. The developed methods, such as Latent Semantic Analysis (LSA) [10, 11], Probabilistic LSA (PLSA) [12] or Latent Dirichlet Allocation (LDA) [2] build a high-level representation of a document in a topic space. Documents are then considered as "bags-of-words" [13] where the word order is not taken into account.

LDA is presented in its plate notation in Figure 2. These methods demonstrated their performance on various tasks, such as sentence [14] or keyword [15] extraction. Contrary to multinomial mixture models, LDA considers that a topic is associated with each occurrence of a word composing the document, rather than with the complete document. Thereby, a document can switch topic at any given word. Word occurrences are connected by a latent variable which controls the global distribution of topics inside a document. These latent topics are characterized by words and their corresponding distribution probability. PLSA and LDA models have been shown to generally outperform LSA on information retrieval tasks [16]. Moreover, LDA provides a direct estimate of the relevance of a topic, given a word set.

The generative process corresponds to the hierarchical Bayesian model shown in Figure 2. Several techniques, such as variational methods [2], expectation-propagation [17] or Gibbs sampling [18], have been proposed to estimate the parameters describing a LDA hidden space. Gibbs sampling is a special case of Markov-chain Monte Carlo (MCMC) [19] and gives a simple algorithm to approximate inference in high-dimensional models such as LDA [20]. This overcomes the difficulty to directly and exactly estimate parameters that maximize the likelihood defined as:

$$P(W|\overrightarrow{\alpha}, \overrightarrow{\beta}) = \prod_{w \in W} P(\overrightarrow{w}|\overrightarrow{\alpha}, \overrightarrow{\beta}) \quad (1)$$

for the whole data collection W given the Dirichlet parameters $\overrightarrow{\alpha}$ and $\overrightarrow{\beta}$.

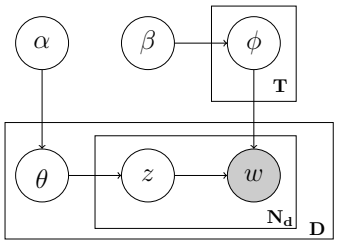


Figure 2: Generative models in plate notation for LDA model.

LDA estimation through Gibbs sampling was firstly reported in [18]; a more detailed description can be found in [20]. This method is used both to estimate the LDA parameters and to infer an unseen document with a hidden space of n topics. According to LDA, topic z is drawn from a multinomial over θ which is drawn itself from a Dirichlet distribution ($\overrightarrow{\alpha}$). In our context, topic space is learned from a lemmatized corpus where each word is associated with its lemma. Thus, a sentence can be inferred from a set of (word, lemma) pairs.

3.2. Topic-based Translation Correction

The first step of the proposed translation correction approach is to spot words that are likely to be mistranslated. For this purpose, a confidence score is computed for each word occurring in a sentence s using n-gram probabilities for each target word computed by the language model. Words with the smallest scores are assumed to be mistranslated and have to be corrected. In this paper, we propose to use a LDA topic space to find out relevant concurrent words w' to replace these suspected mistranslated words w. In order to do so,

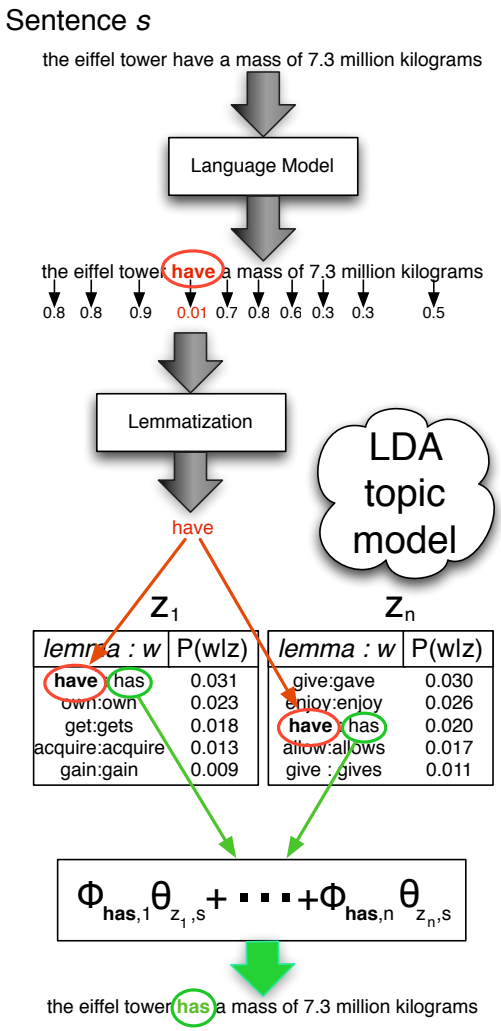


Figure 3: Details about the post-processing correction approach based on a LDA topic space.

Gibbs sampling is used to represent a new sentence s within the topic space of size n ($n = 100$ in our experiments) as shown in Figure 1, and to obtain the topic distribution:

$$\theta_{z_j,s} = P(z_j|s) \ . \tag{2}$$

The next step is to find out a relevant word w' that should replace the erroneous one w. Alternate words are searched among the words having a different inflection but satisfying the constraint:

$$\text{lemma}(w') = \text{lemma}(w) \ .$$

Each topic z is a distribution $P(w|z)$ over the vocabulary. Thus, a thematic confidence score is estimated for a concurrent word w' by:

$$\begin{aligned} \delta(w', s) &= P(w'|s) \\ &= \sum_{j=1}^{n} P(w'|z_j)P(z_j|s) \\ &= \sum_{j=1}^{n} \phi_{w',z_j}\theta_{z_j,s} \end{aligned} \tag{3}$$

where $\phi_{w',z_j} = P(w'|z_j)$ are computed during the training process of the LDA topic space. Each word w' contained in the training corpus is associated with a thematic confidence score δ. Finally, the hypothesis w' with the highest score δ is selected as shown in Figure 3.

4. Experiments

The proposed approach is based on a topic space learned with the LDA MALLET Java implementation[4]. This topic space contains 100 classes and the LDA hyper-parameters are chosen empirically as in [18] ($\alpha = \frac{50}{100} = 0.5$ and $\beta = 0.1$). During the learning process, the MALLET package requires to lowercase input text. For this reason, the results considered for the post-processing step are computed on lowercased sentences.

The effectiveness of the proposed approach is evaluated in the IWSLT benchmark. Table 2 reports case-sensitive BLEU and TER scores measured on the *test0*, *test13* and *test14* corpora, with two factored phrase-based TM model setups: a first one ($w \rightarrow w$) where only words are considered on the source and target sides, and a second one ($w \rightarrow (w, p)$) where English words are translated into (word, PoS) pairs. Disambiguating words with their PoS by the second factored model improves BLEU and TER over the first model for the three test corpora and both studied language pairs. For example, an absolute increase of BLEU (between 0.85 and 1.2) is observed for Slovene; a more limited but still consistent improvement of BLEU (between 0.1 and 0.5) happens for Polish.

Translation produced by the second TM models were used as entry of the LDA post-processing step. Table 3 shows results measured this time in terms of case-insensitive BLEU and TER, since sentences are lowercased before the post-processing step. The thresholds to consider a word as mistranslated from LM-based confidence scores were optimized in terms of BLEU on *test0*. These thresholds lead to change 1.2 % of words for Slovene and around 3 % for Polish (Table 3, columns 3 and 6). Unfortunately, using the proposed LDA-based approach did not translate into an observed gain in terms of BLEU or TER (line 1 vs line 2 and line 3 vs line 4).

[4] http://mallet.cs.umass.edu/

	TM MODELS	*test0*		*test13*		*test14*	
		BLEU	TER	BLEU	TER	BLEU	TER
English → Slovene	$w \rightarrow w$	12.27	69.58	13.20	67.70	10.92	69.66
	$w \rightarrow (w, p)$	13.35	68.64	14.05	66.32	12.16	68.59
English → Polish	$w \rightarrow w$	10.36	77.61	10.78	79.04	9.16	86.68
	$w \rightarrow (w, p)$	10.45	75.70	11.29	76.59	9.63	83.88

Table 2: Case-sensitive BLEU and TER (in %) measured to evaluate the use of a PoS factor inside the TM model.

	TM MODELS	*test0*			*test14*		
		BLEU	TER	% modified words	BLEU	TER	% modified words
English → Slovene	$w \rightarrow (w, p)$	13.68	67.78	-	12.69	67.90	-
	+ post-processing	13.42	68.03	1.16	12.23	68.17	1.29
English → Polish	$w \rightarrow (w, p)$	11.09	74.20	-	10.12	82.51	-
	+ post-processing	10.66	74.95	2.81	9.63	83.39	3.53

Table 3: Case-insensitive BLEU and TER (in %) measured before and after the LDA post-processing step.

5. Conclusions and Perspectives

In this paper, we propose an original post-processing approach to automatically correct translated texts. Our method takes advantage of a latent Dirichlet (LDA) model that provides thematically and grammatically close forms of mistranslated words. Experiments were conducted in the framework of the IWSLT machine translation evaluation campaign on the English-to-Polish and English-to-Slovene tasks. The proposed system was compared to a more classical factored translation system.

Results showed that the original proposed system does not improve results obtained with the baseline one, but we think that this preliminary work should lead to further investigations in the future. For example, we would like to use this model at an early stage, during the decoding process of the MT system, and not only at a post-processing stage. Furthermore, other features than n-gram probabilities should be exploited to identify mistranslated translations [21]. Finally, the low results observed with the topic-based correction approach are obtained with a topic space which still considers sentences as "bag-of-words" and ignore their internal grammatical structure. For this reason, a promising future work is to embed the position of the word in the sentence or n-gram containing the word.

6. References

[1] P. Koehn and H. Hoang, "Factored translation models," in *Proc. of EMNLP-CoNLL*, 2007, pp. 868–876.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[3] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proc. of ACL, Companion Volume*, 2007, pp. 177–180.

[4] M. Grčar, S. Krek, and K. Dobrovoljc, "Obeliks: statistični oblikoskladenjski označevalnik in lematizator za slovenski jezik," in *Proc. of the 15th International Multiconference (IS)*, 2012, pp. 89–94.

[5] H. Schmid, "Improvements in part-of-speech tagging with an application to German," in *Proc. of the ACL SIGDAT Workshop*, 1995, pp. 47–50.

[6] A. Stolcke *et al.*, "SRILM—an extensible language modeling toolkit." in *Proc. of Interspeech*, 2002.

[7] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Proc. of the ACL Workshop: Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, 2008, pp. 49–57.

[8] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of ACL*, vol. 1, 2003.

[9] S. Huet, E. Manishina, and F. Lefèvre, "Factored machine translation systems for Russian-English," in *Proc. of WMT*, 2013.

[10] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[11] J. R. Bellegarda, "A latent semantic analysis framework for large-span language modeling," in *Proc. of Eurospeech*, 1997.

[12] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. of Uncertainty in Artificial Intelligence, UAI ' 99*, 1999.

[13] G. Salton, "Automatic text processing: the transformation," *Analysis and Retrieval of Information by Computer*, 1989.

[14] J. R. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1279–1296, 2000.

[15] Y. Suzuki, F. Fukumoto, and Y. Sekiguchi, "Keyword extraction using term-domain interdependence for dictation of radio news," in *Proc. of Coling*, vol. 2. ACL, 1998, pp. 1272–1276.

[16] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1, pp. 177–196, 2001.

[17] T. Minka and J. Lafferty, "Expectation-propagation for the generative aspect model," in *Proc. of the Conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 352–359.

[18] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National academy of Sciences of the United States of America*, vol. 101, no. Suppl 1, pp. 5228–5235, 2004.

[19] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984.

[20] G. Heinrich, "Parameter estimation for text analysis," Fraunhofer IGD, Tech. Rep., 2009, version 2.9. [Online]. Available: http://www.arbylon.net/publications/text-est.pdf

[21] N. Bach, F. Huang, and Y. Al-Onaizan, "Goodness: A method for measuring machine translation confidence," in *Proc. of ACL*, 2011.

The USFD SLT system for IWSLT 2014

Raymond W. M. Ng, Mortaza Doulaty, Rama Doddipatla, Wilker Aziz, Kashif Shah, Oscar Saz, Madina Hasan, Ghada AlHarbi, Lucia Specia, Thomas Hain

Department of Computer Science, University of Sheffield, United Kingdom

{wm.ng,mortaza.doualty,r.doddipatla,w.aziz,kashif.shah, o.saztorralba,m.hasan,GAlHarbi1,l.specia,t.hain}@sheffield.ac.uk

Abstract

The University of Sheffield (USFD) participated in the International Workshop for Spoken Language Translation (IWSLT) in 2014. In this paper, we will introduce the USFD SLT system for IWSLT. Automatic speech recognition (ASR) is achieved by two multi-pass deep neural network systems with adaptation and rescoring techniques. Machine translation (MT) is achieved by a phrase-based system. The USFD primary system incorporates state-of-the-art ASR and MT techniques and gives a BLEU score of 23.45 and 14.75 on the English-to-French and English-to-German speech-to-text translation task with the IWSLT 2014 data. The USFD contrastive systems explore the integration of ASR and MT by using a quality estimation system to rescore the ASR outputs, optimising towards better translation. This gives a further 0.54 and 0.26 BLEU improvement respectively on the IWSLT 2012 and 2014 evaluation data.

1. Introduction

In this paper, the University of Sheffield (USFD) system for the International Workshop on Spoken Language Translation (IWSLT) 2014 is introduced. USFD participated in English-to-French and English-to-German SLT tasks. The ASR and MT systems made use of state-of-the-art technologies. On the ASR side, two deep neural network systems built on partially different data and different tandem configurations were used. On the MT side, phrase-based translation models were built. ASR and MT system integration attempts were made by using a translation quality estimation system. It considered the system scores from both ASR and MT, as well as features extracted from the ASR outputs in source language. The ASR hypotheses were then rescored based on the predicted translation quality. This gives performance improvements in terms of BLEU score increase.

In the following, the data used for system training is introduced in §2. §3 and §4 give the details of the ASR and MT systems. The decoding algorithm and system results are given in §5. Besides the primary submission, USFD also submitted contrastive systems which implement system integration. These systems used a quality estimation module and performed ASR N-best list rescoring based on predicted translation quality. This would be described in §6.

2. Data processing and selection

The ASR and MT systems were primarily trained on TED lecture data [1]. For ASR, TED and the additional data form two data subsets, on which two systems were trained. For MT, out-of-domain data after data selection were incorporated in the training of translation models and target language models.

2.1. ASR acoustic modelling

Two data sets were used for ASR system training. For the ease of discussion they are hereinafter referred to as ASR_1 and ASR_2. The composition of the two data sets is shown in Table 1.

Table 1: Data for acoustic model training

ASR_1		ASR_2	
Data	Hours	Data	Hours
TED	132	TED	112
LLC	106	AMI+AMIDA+ICSI	165
ECRN	60	ECRN	60

TED serves as a common data set in both ASR_1 and ASR_2. Their segmentations in ASR_1 and ASR_2 differ slightly and this is explained later. The two data sets are augmented by e-corner lecture data (ECRN) with a duration of 60 hours [2]. ASR_1 also contains 106 hours of LLC lecture data. In ASR_2, 165 hours of meeting data from the AMI, AMIDA and ICSI corpora are added so the trained model will reflect also generic domains other than lectures [3, 4].

The TED portions in both ASR_1 and ASR_2 originate from 734 TED talks published before 31 Dec 2010. Each talk has a duration of around 15 minutes. Human annotations in the form of subtitles are also available, giving rough segmentation with segment duration from 3 to 5 seconds and time accuracy to the nearest second.

Exact segmentations and transcriptions of TED were derived in different ways in ASR_1 and ASR_2. In ASR_1, all segments from the same talk were merged and the speech was forced aligned, resegmented before another forced alignment run determined the final training set. This gave a total of 132 hours of speech for AM training. In ASR_2, forced alignment

Table 2: Amount of text data used in different training tasks in En→Fr translation (#Full data set was used for builing target LM)

	Number of words/million			
Data	Target LM#	Source LM	Punct TM	TM
TED	3.17	3.17	3.17	3.17
News Commentary	4.0	0.9	0.2	0.7
Common crawl	70.7	36.1	3.6	10.8
Gigaword	575.7	271.2	26.3	14.9
Europarl	50.3	10.8	4.3	1.9

was performed on the rough segmentation, after which contagious segments were merged when there was tight silence at the segment boundaries. A further run of forced alignment determined the final training set. This gave a total of 112 hours of speech.

To evaluate the performance of different segmentations, PLP-based state-tied triphone models with cepstral mean and variance normalisation were trained on these data and decoding was performed on the IWSLT 2010 evaluation data set. The WERs for the ASR_1 and ASR_2 settings are 25.7% and 26.2% respectively. When the models are trained directly on the roughly segmented data (no adjustment of segmentations), the total duration of training data is 109 hours and the corresponding WER is 28.1%.

2.2. Language models and MT

Textual data for the training of language models and translation models were obtained from the affiliated websites of the IWSLT and WMT evaluations [5, 6]. TED was considered as the in-domain training data and the full data set was used. Four out-of-domain (OOD) data sets from News commentary v9, Common Crawl, Gigaword and Europarl v7 were also used, after a data selection process.

The OOD corpora were selected with the cross entropy difference criterion [7]. Given a sentence $x_1^I = [x_1 \cdots x_I]$ with I words, cross entropy values $H(x_1^I, ID)$ and $H(x_1^I, OOD)$ were computed using $\mathcal{G}_{\text{ID}}$, the ID language model (in this case, TED) and $\mathcal{G}_{\text{OOD}}$, the OOD language model (built on the corpus from which the sentence was taken). The cross entropy difference (CED) was given by,

$$\text{CED}(x_1^I) = H(x_1^I, \mathcal{G}_{\text{ID}}) - H(x_1^I, \mathcal{G}_{\text{OOD}}) \qquad (1)$$

Sentences were ranked by the CED values and 25% of the sentences with the lowest CED values were selected from each corpus. Furthermore, CED values were calculated on sentence batches with increasing sizes. A line search was done to find the optimal batch giving the minimum CED value. All data selection was done on the English text. For data selection to translation model training, the corresponding sentences in the target languages were extracted after selection was done on English sentences.

Table 2 shows the amount of the full text data set, and the selected text data in different systems in the English→French translation task. The full data set contains 703.9M words. They were used for training the target language model in MT, which was a 5-gram interpolated LM with punctuation and out-of-vocabulary word modelling, modified Kneser-Ney smoothing and was in standard ARPA format. The source language model for ASR was built on the full TED data set and 25% or 50% of the OOD data, making up to 322.2M words. A monolingual translation model was trained for punctuation insertion and case conversion. The training took the full TED data and 5-10% of the OOD data, resulting in a total of 37.6M words. The translation model was trained on the full TED data set and other optimally selected OOD data sets, where only around 5% of the sentences were selected. The total number of words is 31.7M.

3. Automatic speech recognition

There are two DNN systems with tandem configurations in ASR [8]. Bottleneck (BN) features were derived from deep neural network (DNN)s [4], and GMM-HMM systems were trained on these bottleneck features. The two tandem systems were trained on ASR_1 and ASR_2 data respectively (Table 1). Different portions of data were used in different stages of training. Let DNN_1 and DNN_2 denote the two DNN systems for ASR_1 and ASR_2. DNN_1 was trained on TED data only. DNN_2 was trained on TED and AMI+AMIDA+ICSI data only. The remaining data listed in Table 1 were added to the training pool in the GMM-HMM training stage.

DNN_1 has 4 hidden layers, each having 1,745 hidden units. The BN layer is placed just before the output layer and has 26 units. The output layer has 4,320 units. DNN_2 has 5 hidden layers, with the first 3 layers having 1,745 units and the fourth hidden layer having 65 units. A BN layer is placed just before the output layer and has 39 units. The output layer has 5,691 units.

Both the DNNs were trained using log filter-bank outputs and concatenating 31 adjacent frames, which were decorrelated using DCT to form a 368-dimensional feature vector. The filter-bank outputs were mean and variance normalised at the speaker level. Global mean and variance normalisation was performed on each dimension before feeding the input for training the DNN. The GMM-HMM systems trained using the BN features were different. The model for ASR_1 was trained on the concatenated features with the 26-dimension BN features from DNN_1 and the 39-dimension PLP features. The model for ASR_2 was trained on the 39-dimension BN features from DNN_2. Both the GMM-HMM models were trained as tied-state triphone systems with the final models having 16 mixture Gaussians per state.

All systems are vocal tract length normalised (VTLN). In the training stage, a PLP system was used to obtain the warp factors for each speaker. Then the filter-bank and PLP features were VTLN-warped, which were in turn used for DNN and GMM-HMM training in the tandem configuration. In the decoding stage, a non-VTLN DNN and GMM-HMM tandem

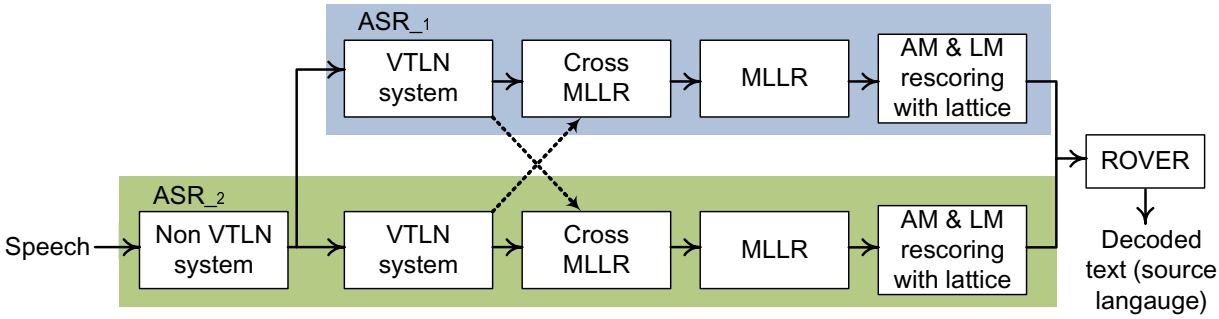


Figure 1: System diagram for multi-pass ASR decoding.

system trained on ASR_2 data replaced the PLP system for the derivation of warp factors.

To improve the performance of the acoustic model, minimum phone error (MPE) training was performed using the lattices which were generated using a uni-gram language model [9].

Language models for ASR are all interpolated LMs built on the English text data described in Table 2 and tuned on IWSLT 2010 dev and eval data. 2-gram and 4-gram ARPA language models were trained for lattice generation and expansion. The 4-gram LM was pruned with a threshold 10^{-10} and a weighted-finite-state transducer (WFST) was constructed for fast decoding in the pre-final passes in the ASR systems.

All ASR LMs were based on a word-list with a 60k word vocabulary extracted based on our standard English ASR inventory and the English part of the TED MT training data for IWSLT 2014 [3, 5]. Pilot ASR experiments on the IWSLT 2011 and 2012 eval data show the drop of perplexity with the addition of Common crawl and Gigaword data. For these two corpora, the rate of data selected for LM building was set to 50%, while the rate for other OOD corpora was kept 25%. This made the total number of words 322.2M as shown in Table 2.

Pronunciation probabilities were incorporated in final stage decoding [10]. These probabilities were extracted based on the Viterbi alignment of the phoneme level transcription of the ASR_1 training data. When a word allowed multiple pronunciations, the frequency of each pronunciation was calculated and stored. These frequencies were then applied to the words in the decoding dictionary for words that appeared in both training and decoding stages. Words with multiple pronunciations appearing only in the decoding stage were given equal probability.

4. Machine translation

A phrase-based model using MOSES [11] in a standard setting was employed. For phrase extraction all of the TED data (3.17 million words) was used. Following previous findings [12], data selection via a cross-entropy difference criterion (detailed in §2.2) was used to select the optimal batch of the OOD data, which amounts to about 5% of the total data or 30.58M words. The phrase length was limited to 5 and word-alignment was obtained with FASTALIGN [13]. Lexicalised reordering models were trained using the same data. For language modelling, we used the complete sets of OOD data (i.e. no data selection). 5-gram LMs were trained using LMPLZ [14]. 100-best MIRA tuning was employed [15]. For the English-to-French system, tuning was done on the IWSLT 2010 development and evaluation data with a total of 2,551 sentences. For the English-to-German system, tuning was done on the IWSLT 2010 development data with 887 sentences.

In SLT, the input to the MT system was ASR output, which typically lacks casing and punctuation. Following previous work [16, 17], a monolingual translation system was trained to recover casing and punctuation from the ASR output, thus producing source sentences which are more adequate for translation. The training data for this monolingual MT system was obtained by pre-processing an actual corpus of the source language to form *pseudo ASR* outputs, which contained no case and punctuation information. Numbers, symbols and acronyms were also converted to their verbal forms with lookup tables. We then used this synthesised corpus of pseudo ASR as the source, and the original corpus as the target of our monolingual MT. The monolingual translation system was trained on 37.6M words (Table 2). It performed monotonic translation with phrases of as long as 7 words.

5. Decoding

The evaluation systems for ASR and MT are multi-pass systems with resource optimisation and environment management capabilities [11, 18]. The ASR is a two-stream multi-pass system. It is illustrated in Figure 1. The two streams ASR_1 and ASR_2 differ by the acoustic model training data (detailed in Table 1) and also the tandem configurations (detailed in §3). Both streams follow the same routine along the multi-pass decoding system. In pass 1, a unified decoding result was generated using a non-VTLN DNN and GMM-HMM tandem system with cepstral mean and variance (CMVN) normalisation trained on ASR_2 data. These

Table 3: Tree-search and WFST decoder

	Tst11		Tst12	
Decoder	WER	RT	WER	RT
Tree-search	23.7%	18.4	27.0%	19.8
WFST	23.7%	3.0	27.0%	3.3

hypothesis transcripts were used for inferring the warp factors. The filterbank (for both ASR_1 and ASR_2) and PLP (for ASR_1 only) features were then warped and CMVN normalised, and the system branched off into two streams with two VTLN decoders trained on ASR_1 and ASR_2 data respectively.

After pass 2 decoding, speaker-based MLLR cross adaptations were carried out. The transcripts from ASR_1 was used for the model transformation in ASR_2 system and vice versa. The number of regression classes was set to 16. When pass 3 decoding was done, MLLR self adaptations were performed. The number of regression classes was also set to 16.

All pre-final stage decoding made use of weighted finite state transducers (WFSTs) for fast implementation. In a pilot experiment, PLP systems with heteroscedastic linear discriminant analysis (HLDA) were trained on the ASR_2 data [19]. WFST decoding with a pruned 4-gram grammar network was compared with the standard tree search with an unpruned 3-gram LM. The WER and real-time factor (RT) on IWSLT 2011 evaluation and IWSLT 2012 evaluation data are shown in Table 3. WFST was shown to achieve the same performance as tree-search decoding, with much faster decoding speed.

In the final stage, acoustic and language model rescoring were performed. Base lattices were generated with 2-gram LM pruned with a threshold 10^{-10}. Lattice expansion was done with 4-gram unpruned language models. Three settings were tried and the results were compared,

(i) Language model rescoring with the 4-gram LM
(ii) Considering pronunciation probability (Pron. prob.) on top of (i)
(iii) Acoustic and language model rescoring with the setting of (ii)

ASR performance in terms of WER are shown in Table 4. The initial non-VTLN system gave WER of 16.9% and 17.7% on IWSLT 2011 and 2012 data respectively. Moving towards the VTLN systems, when ASR_1 and ASR_2 branched off, it is observed that the ASR_1 model gave 1.0% to 1.4% lower WER than the ASR_2 model. This is because the data in ASR_1 had a better match in terms of domain. Incremental performance gains can be observed in individual steps, particularly MPE, cross-adaptation and language model rescoring. The WER difference between ASR_1 and ASR_2 diminished to 0.4-0.5% after all optimisation steps. After system combination, the final WER is 21-25% relatively lower compared with the initial system.

MT Decoding was performed with cube pruning [20] both in tuning and testing. Decoding was done with the minimum Bayes risk criterion and reordering over punctuations was forbidden. To restore the correct case of the output the truecasing heuristic was employed. The same set of standard techniques was applied on En→Fr and En→De translation.

Table 4: WER of the multi-pass ASR systems

	Tst11		Tst12	
ASR system	ASR_1	ASR_2	ASR_1	ASR_2
Non-VTLN	–	16.9%	–	17.7%
+VTLN	15.4%	16.4%	16.4%	16.8%
+MPE	14.7%	15.7%	16.0%	16.1%
+Cross-adapt	14.0%	14.9%	14.2%	14.8%
+Self-adapt	14.0%	15.0%	14.2%	14.7%
+LM rescoring	13.4%	14.5%	13.5%	14.2%
+Pron. prob.	13.3%	14.2%	13.4%	14.0%
+AM rescoring	13.3%	13.8%	13.4%	13.7%
ROVER	—13.3%—		—13.2%—	

Table 5: MT system performance on eval data

	BLEU(c)	
Language pair	Dev10	Tst12
(MT with true transcript)		
En→Fr		40.9
En→De	21.5	
(Monolingual translation)		
En(pseudo ASR)→En		88.0
En(ASR)→En		69.0
(SLT)		
En(ASR)→En→Fr		31.7
En(ASR)→En→De	16.8	

The MT system was tested on IWSLT 2010 development data and 2012 evaluation data, and the results are shown in Table 5. Performance are shown in terms of cased and punctuated BLEU scores. When given the reference transcript, the MT system gave 40.9 and 21.5 BLEU score for MT tasks in En→Fr and En→De respectively. The monolingual translation system (§4) restored case and punctuation information. It was tested on pseudo ASR and real ASR output and yielded 88.0 and 69.0 BLEU score. Finally in the SLT setting, the decoded ASR result was fed to the monolingual translation system and the output were subsequently translated. The BLEU score is 31.7 and 16.8 for SLT tasks in En→Fr and En→De respectively.

In Table 6, the official IWSLT 2014 evaluation performance in terms of BLEU and TER (cased, punctuated and non-case, non-punctuated) for the USFD primary system is shown.

Table 6: Primary SLT system performance (Tst14)

Language pair	BLEU(c)	TER(c)	BLEU	TER
En→Fr	23.45	59.94	24.14	58.97
En→De	14.75	70.15	15.24	69.15

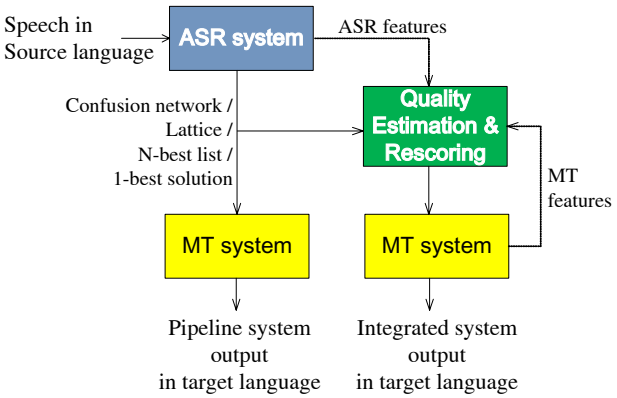


Figure 2: System integration with ASR and MT

6. System integration

The USFD primary system is a pipeline SLT system in which 1-best ASR result was directly fed to the MT system. System integration experiments were tried in the En→Fr SLT task and the results were submitted as contrastive systems. Figure 2 depicts the integrated system and its comparison with the pipeline system. In the integrated system, ASR system hypotheses are expanded in the form of lattices, confusion networks or N-best lists. A quality estimation (QE) module evaluated and rescored the ASR outputs before they were fed to the MT system.

In our implementation, 10-best outputs from the ASR system on the IWSLT 2011 evaluation data were used for QE training. The QE module derived 117 QuEst [21, 22] features from each sentence to describe its linguistic, statistical properties as well as the statistics from the ASR and MT models. Out of the 117 features, top 58 features were selected using the Gaussian Process (GP) with RBF kernel as described in [23]. Further, GP was used to learn the relationship between the selected features and the translation performance of the sentence (in this case, sentence-based METEOR score) [24]. During testing, the estimated translation performance was used to rescore the 10-best ASR output. Details of the integrated system were described in [25].

Table 7: Contrastive SLT system performance (En→Fr)

Setting	Tst12	Tst14
Contrastive 1 (baseline)	31.33	23.18
Contrastive 2 (+ 10-best list rescoring)	31.51	23.27
Contrastive 3 (+ ASR confidence-informed rescoring)	31.87	23.44

The ROVER combination of ASR_1 and ASR_2 systems only provided 1-best output. In the integration experiment, the 10-best output from ASR_1 was used instead.

Performance of the contrastive systems in terms of cased and punctuated BLEU score is shown in Table 7. Contrastive 1 result is from the baseline system with pipeline setting. Contrastive 2 and 3 show the results of two different system integration settings. The baseline system gave BLEU scores 31.33 and 23.18 on IWSLT 2012 and IWSLT 2014 data. The baseline numbers are inferior to the primary system number (IWSLT 2012: 31.7; IWSLT 2014: 23.45) as shown in Table 5 and 6. This is because the baseline here did not benefit from ASR system combination.

Rescoring gives 0.18 and 0.09 BLEU improvements to IWSLT 2012 and IWSLT 2014 data respectively. By inspecting the results, it was found that rescoring generally had higher effectiveness for the sentences with low ASR confidence. Therefore, a confidence threshold was set, and rescoring was only performed when the ASR confidence dropped below this threshold. For IWSLT 2012 data, optimality was reached when 55% of the sentences were selected by this confidence criteria to rescore, resulting a further 0.36 BLEU score gain. This threshold was applied on IWSLT 2014 data, a 0.17 BLEU score gain was observed.

7. Summary

In this paper, the USFD SLT system for IWSLT 2014 was described. Automatic speech recognition (ASR) is achieved by two multi-pass deep neural network systems with slightly different tandem configurations and different training data. Machine translation (MT) is achieved by a monolingual phrase-based monotonic translation system which recovers case and inserts punctuation, followed by a bilingual phrase-based translation system. The USFD contrastive systems explore the integration of ASR and MT by using a quality estimation system to rescore the ASR outputs, optimising towards better translation. This gives noticeable BLEU improvement on the IWSLT 2012 and 2014 evaluation data.

8. References

[1] TED, "Technology entertainment design," http://www.ted.com, 2006.

[2] M. Hasan, R. Doddipatla, and T. Hain, "Multi-pass sentence-end detection of lecture speech," in *Proc. Interspeech*, 2014.

[3] T. Hain, L. Burget, J. Dines, P. N. Garner, A. E. Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "The AMIDA 2009 meeting transcription system," in *Proc. Interspeech 2010*, 2010, pp. 358–361.

[4] R. Doddipatla, M. Hasan, and T. Hain, "Speaker dependent bottleneck layer training for speaker adaptation in automatic speech recognition," 2014.

[5] M. Cettolo, C. Girardi, and M. Federico, "WIT3: Web inventory of transcribed and translated talks," in *Proceedings of Conference of European Association for Machine Translation Trento (Italy)*, 2012, pp. 261–268.

[6] "ACL 2014 ninth workshop on statistical machine translation," http://www.statmt.org/wmt14/translation-task.html, 2014.

[7] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the ACL 2010 Conference Short Papers*, ser. ACLShort '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 220–224.

[8] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000.

[9] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP*, 2002.

[10] T. Hain, "Implicit modelling of pronunciation variation in automatic speech recognition," *Speech Communication*, vol. 46, pp. 171–188, 2005.

[11] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 177–180.

[12] A. Birch, N. Durrani, and P. Koehn, "Edinburgh SLT and MT system description for the IWSLT 2013 evaluation," in *Proceedings of International Workshop on Spoken Language Translation*, 2013.

[13] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 644–648.

[14] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable modified Kneser-Ney language model estimation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 690–696.

[15] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ser. NAACL HLT '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 427–436.

[16] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modelling punctuation prediction as machine translation," in *Proc. IWSLT*, 2011.

[17] E. Cho, J. Niehues, and A. Waibel, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system," in *Proc. IWSLT*, 2012.

[18] T. Hain, A. E. Hannani, S. N. Wrigley, and V. Wan, "Automatic speech recognition for scientific purposes - webASR," in *Proc. Interspeech*, 2008, pp. 504–507.

[19] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Communication*, vol. 26, pp. 283–297, 1998.

[20] L. Huang and D. Chiang, "Forest rescoring: Faster decoding with integrated language models," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 144–151.

[21] L. Specia, K. Shah, J. G. C. d. Souza, and T. Cohn, "QuEst - A translation quality estimation framework," in *Proceedings of 51st Annual Meeting of the Association for Computational Linguistics: Demo Session*, Sofia, Bulgaria, 2013, p. 794.

[22] K. Shah, E. Avramidis, E. Biçici, and L. Specia, "QuEst - design, implementation and extensions of a framework for machine translation quality estimation," *Prague Bull. Math. Linguistics*, vol. 100, pp. 19–30, 2013.

[23] K. Shah, T. Cohn, and L. Specia, "An Investigation on the Effectiveness of Features for Translation Quality Estimation," in *Machine Translation Summit XIV*, Nice, France, 2013, pp. 167–174.

[24] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of WMT14*, 2014.

[25] R. W. M. Ng, K. Shah, W. Aziz, L. Specia, and T. Hain, "Quality estimation for ASR K-best list rescoring in spoken language translation," Submitted to *Proc. ICASSP*, 2015.

The Speech Recognition Systems of IOIT for IWSLT 2014

Quoc Bao Nguyen[1], *Tat Thang Vu*[2], *Chi Mai Luong*[2]

[1] University of Information and Communication Technology, Thai Nguyen University, Vietnam
[2] Institute of Information and Technology (IOIT), Vietnamese Academy of Science and Technology (VAST)
nqbao@ictu.edu.vn, {vtthang,lcmai}@ioit.ac.vn

Abstract

This paper describes the speech recognition systems of IOIT for IWSLT 2014 TED ASR track. This year, we focus on improving acoustic model for the systems using two main approaches of deep neural network which are hybrid and bottleneck feature systems. These two subsystems are combined using lattice Minimum Bayes-Risk decoding. On the 2013 evaluations set, which serves as a progress test set, we were able to reduce the word error rate of our transcription systems from 27.2% to 24.0%, a relative reduction of 11.7%.

1. Introduction

The International Workshop on Spoken Language Translation (IWSLT) is a yearly scientific workshop, associated with an open evaluation campaign on spoken language translation. One part of the campaign focuses on the translation of TED Talks, which are a collection of public lectures on a variety of topics, ranging from Technology, Entertainment to Design. As in the previous years, the evaluation offers specific tracks for all the core technologies involved in spoken language translation, namely automatic speech recognition (ASR), machine translation (MT), and spoken language translation (SLT).

The goal of the ASR track is the transcription of audio coming from unsegmented TED and TEDx talks, in order to interface with the machine translation components in the speech-translation track. The quality of the resulting transcriptions are measured in word error rate (WER).

In this paper we describe the speech recognition systems which we participated in the TED ASR track of the 2014 IWSLT evaluation campaign. This year, our system is a further development of our last year's evaluation system [1], and focuses on improving acoustic model using deep neural network. There are two main approaches for incorporating artificial neural networks in acoustic modeling today: hybrid systems and tandem systems. In the hybrid approach, a neural network is trained to estimate the emission probabilities for Hidden Markov Models (HMM) [2]. In contrast, tandem systems use neural networks to generate discriminative features as input values for the common combination of Gaussian Mixture Models (GMM) and HMMs. One of the common tandem system uses the activations of a small hidden layer ("bottleneck features", BNF [3]).

The organization of the paper is as follows. Section 2 describes the data that our system was trained on. This is followed by Section 3 which provides a description of the way to extract deep bottleneck features. An overview of the techniques used to build our acoustic models is given in Section 4. Dictionary and language model are presented in Section 5. We describe the automatic segmentation process in Section 6. Our decoding procedure and results are presented in Section 7.

2. Training Corpus

For acoustic model training, we used TED talk lectures (http://www.ted.com/talks) as training data. Approximately 220 hours of audio, distributed among 920 talks, were crawled with their subtitles, which were deliberately used for making transcripts. However, the provided subtitles do not contain the correct time stamps corresponding with each phrase as well as the exact pronunciation for the words spoken, which lead to the necessity for long-speech alignment.

Segmenting the TED data into sentence-like units used for building a training set was performed with the help of SailAlign tool [4] which helps us to not only acquire the transcript with exact timing, but also to filter non-spoken sounds such as music or applauses. A part of these noises are kept for noise training while most of them are abolished. After that, the remained audio used for training consists of around 160 hours of speech.

3. Deep Bottleneck Features

In this work, we applied the deep neural network architecture for bottleneck feature extraction (DBNFs) as in [5] [6] and depicted in Figure 1. The network consists of a variable number of moderately large, fully connected hidden layers and a small bottleneck layer which is followed by an additional hidden layer and the final classification layer.

The Mel-frequency cepstral coefficients (MFCCs) features were used as input of deep neural network, which contain 39 coefficients including 12 cepstral coefficients, 1 energy coefficient added with delta and double-delta features were extracted after windowing with the window size of 25 milliseconds and frame shift of 10 milliseconds. Then they were pre-processed using the approach in [7] that is called

splicing speaker-adapted features with 40 dimensions. This features for each frame were stacked with 9 adjacent samples, resulting in a total of 360 dimensions. For pre-training the stack of auto-encoders, mini-batch gradient descent with a batch size of 128 and a learning rate of 0.01 was used. Input vectors were corrupted by applying masking noise to set a random 20% of their elements to zero. Each auto-encoder contained 1024 hidden units and received 1 million updates before its weights were fixed and the next one was trained on top of it.

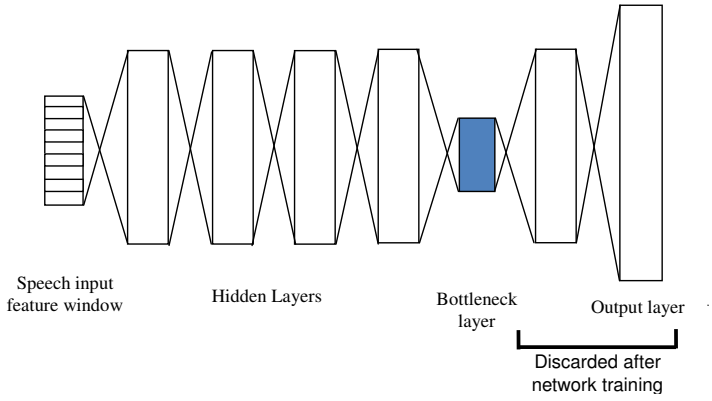


Figure 1: Deep Network Architecture for Bottleneck Features

The remaining layers were then added to the network, with the bottleneck layer consisting of 39 units, another hidden layer and output layer containing 4,500 context-dependent HMM states. Again, gradients were computed by averaging across a mini-batch of training examples; for fine-tuning, we used a larger batch size of 256. The learning rate was decided by the newbob schedule: for the first epoch, we used 0.008 as the learning rate, and this was kept fixed as long as the increment in cross-validation frame accuracy in a single epoch was higher than 0.5%. For the subsequent epochs, the learning rate was halved; this was repeated until the increase in cross-validation accuracy per epoch is less than a stopping threshold, of 0.1%. The activations of the 39 bottleneck units are stacked over an 9-frame context window and reduced to a dimensionality of 40 using linear discriminant analysis (LDA) and maximum likelihood linear transformation (MLLT).

4. Acoustic Model

4.1. Baseline Acoustic Model

Baseline HMM/GMM acoustic model were performed with the Kaldi developed at Johns Hopkins University [8]. Nine consecutive MFCC feature frames were spliced to 40 dimensions using linear discriminant analysis (LDA) and maximum likelihood linear transformation (MLLT) that is a feature orthogonalizing transform, was applied to make the features more accurately modeled by diagonal-covariance Gaussians.

All models used 4,500 context-dependent state and 96,000 Gaussian mixture components. The baseline systems were built, follow a typical maximum likelihood acoustic training recipe, beginning with a flat-start initialization of context-independent phonetic HMMs, followed by tri-phone system with 13-dimensional MFCCs plus its deltas and double-deltas and ending with tri-phone system and LDA+MLLT.

4.2. Hybrid Acoustic Model

For the hybrid network training, we used the same techniques that described in the deep bottleneck feature section. The network architecture, we settled with a stacked 5 auto-encoders containing 1024 units each. Its input was used the same with DBNFs network, MFCCs feature were pre-processed and stacked over a 9 adjacent frames. 4,500 context-dependent target states were used for supervised training that is the number of tied states in the respective baseline systems.

5. Dictionary and Language Model

The word set contains 131,137 words. The lexicon was built based on the Carnegie Mellon University (CMU) Pronouncing Dictionary v0.7a; the phoneme set contains 39 phonemes. This phoneme (or more accurately, phone) set is based on the ARPAbet symbol set developed for speech recognition uses. The vowels may carry lexical stress, ranging from no stress, primary stress to secondary stress.

For language modeling, the in-domain data was provided by organizer and 1/8 of Giga corpus was also utilized by filtering it according to the Moore-Lewis approach [9]. Both two datasets were normalized using the normalization toolkit from CMU. The vocabulary used to train language models is the same as in the lexicon. The training data for language model is summarized in Table 1.

Table 1: *Training data for language modeling for English ASR Task.*

Data	Number of sentences	Number of tokens
TED	156,460	2,708,816
1/8 Giga	2,565,687	56,488,064

We trained 3-gram language model using SRILM toolkit with the modified interpolated Knesey-Ney smoothing technique [10] from each of data set. These were then combined using linear interpolation as follows:

$$P(w|h) = \lambda_1 P_1(w|h) + \lambda_2 P_2(w|h) + ... + \lambda_n P_n(w|h)$$

Where $\lambda_1, \lambda_2, ..., \lambda_n$ are the interpolation weights which were chosen to maximize the likelihood of a held out TED data set.

6. Auto Segmentation

The evaluation data has only provided unsegmented audio data since last year. Therefore, in our works the LIUM Diarization toolkit [11] was used to divide the talk into small sentence-like segments. Figure 2 provides a general description on the diarization process. First, 13 MFCC features were extracted from the long audio file. After that, a Viterbi decoding is performed to generate a segmentation. Some of segment boundaries produced by the Viterbi decoding fall within words. These boundaries are adjusted by applying a set of rules defined experimentally. Detection of gender and bandwidth is then done using a GMM for each of the 4 combinations of gender (male / female) and bandwidth (narrow / wide band). Finally, GMMbased speaker clustering is carried out to map each speech segment to the corresponding speaker.

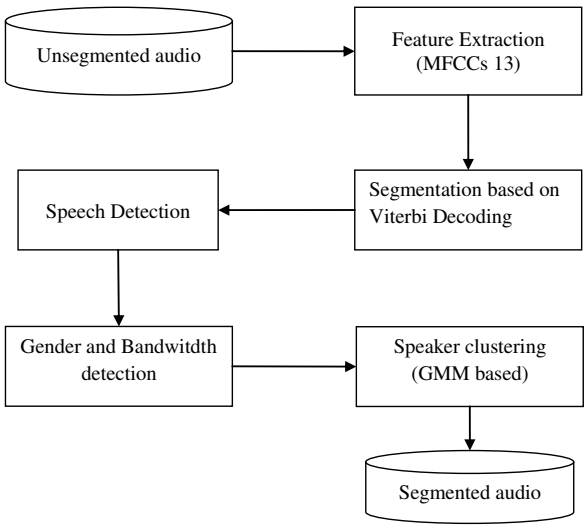


Figure 2: Deep Network Architecture for Bottleneck Features

Comparing automatic and manual segmentation, the disparity in word error rates is disclosed in Table 2. It is notable that the automatic speech detection caused approximately 2 percent loss of the spoken audio, resulted in inevitably decreasing the error rates, presented by deletions. Experiments conducted with tst2013 data illustrated that the WER increased 10% relatively, compared with the same data sets which are manually segmented. The segmentation cannot be guaranteed to be precise at the beginning or the end of the sentence, the output segments are sometime incomplete sentence, or incomplete phrases, which affects recognition results. Last year, we proposed a type of recurrent neural network language model(RNNLM) [1] to resolve this problem. We did not use RNNLM this year because of time consuming.

7. Decoding Procedure and Results

During development, we evaluated our system using the 2012 development set and 2013 test set that released by the IWSLT organizers.

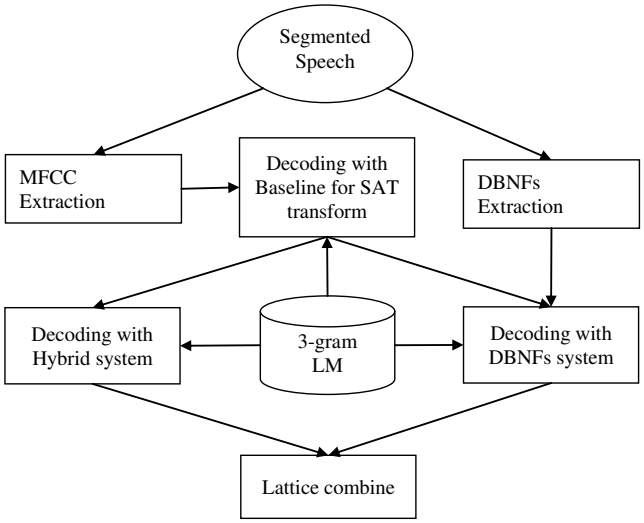


Figure 3: The full decoder architecture

Figure 3 shows the complete decoding architecture. After feature extraction step, followed by decoding with the baseline system to estimate the transformations for speaker adaptation (fMLLR algorithm), we operate two parallel decoding sequences for the tandem and hybrid acoustic models. For each model, the complete process consists of a decoding with the trigram LM using Kaldi decoder tool. Lattices output from the this pass were combined using Lattice Minimum Bayes-Risk (MBR) decoding as described in [12]

Table 2: *English ASR results for various acoustic models and segmentation types (manual, auto)*

System	**WER(%)**		
	dev2012	**tst2013**	**tst2013 auto**
Baseline	30.0	36.1	–
Last year	22.9	29.5	27.2
DBNFs(S1)	19.5	23.8	25.7
Hybrid(S2)	20.0	23.6	25.3
S1+S2	18.7	22.7	24.0

Table 2 lists the performance of our systems in terms of the word error rate (WER). Regarding the performance of the baseline system, the WER is 30.0% on dev2012 and 36.1% on tst2013. The second row is the number of the best system from last year [1] where we applied state-of-the-art techniques for acoustic model without deep neural network. Results for applying deep bottleneck features are listed on third

row of the table. As we can see the results, the DBNF numbers are about 10% absolute (about 33% relative) better than the baseline numbers on both sets. The hybrid DNN/HMM combination also outperforms baseline setup with similar results to DBNFs number. The last row on the table shows the final system combination results of DBNFs and Hybrid systems that gives a further 1% absolute WER reduction as compared to the best single system.

8. Conclusions

In this paper, we presented our English LVCSR systems, with which we participated in the 2014 IWSLT evaluation. The acoustic model was improved using deep neural network for this year evaluation. On the 2012 development set for the IWSLT lecture task our system achieves a WER of 18.7%, and a WER of 24.0% on the 2013 test set.

In the future, we intend to improve language model using deep neural network as in [1] as well as apply a hybrid DNN on top of deep bottleneck features [6] and multi-lingual network training approaches [13] to improve acoustic model for the systems.

9. Acknowledgements

This work was partially supported by Project: "Development of spoken electronics newspaper system based on Vietnamese text-to-speech and web-based technology", VAST01.02/14-15

10. References

[1] N. Q. Pham, H. S. Le, T. T. Vu, , and C. M. Luong, "The speech recognition and machine translation system of ioit for iwslt 2013," in *Proceedings of the International Workshop for Spoken Language Translation (IWSLT)*, 2013.

[2] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Norwell, MA, USA: Kluwer Academic Publishers, 1993.

[3] F. Grezl, M. Karafiat, S. Kontair, and J. Cernocky, "Probabilistic and bottle-neck features for lvcsr of meetings," in *Acoustics, Speech and Signal Processing (ICASSP), 2007 IEEE International Conference on. IEEE*, 2007, pp. V–757 – IV–760.

[4] A. Katsamanis, M. Black, P. G. Georgiou, L. Goldstein, and S. S. Narayanan, "Sailalign: Robust long speech-text alignment," in *Proc. of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, jan 2011.

[5] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *ICASSP2013*, Vancouver, CA, 2013, pp. 3377–3381.

[6] Q. B. Nguyen, J. Gehring, K. Kilgour, and A. Waibel, "Optimizing deep bottleneck feature extraction," in *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2013 IEEE RIVF International Conference on*, Nov 2013, pp. 152–156.

[7] S. P. Rath, D. Povey, K. Vesely, and J. Cernocky, "Improved feature processing for deep neural networks." in *INTERSPEECH*. ISCA, 2013, pp. 109–113.

[8] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[9] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the ACL 2010 Conference Short Papers*, ser. ACLShort '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 220–224.

[10] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ser. ACL '96. Stroudsburg, PA, USA: Association for Computational Linguistics, 1996, pp. 310–318.

[11] S. Meignier and T. Merlin, "Lium spkdiarization: an open source toolkit for diarization," in *in CMU SPUD Workshop*, 2010.

[12] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech & Language*, vol. 25, no. 4, pp. 802–828, 2011.

[13] Q. B. Nguyen, J. Gehring, M. Muller, S. Stuker, and A. Waibel, "Multilingual shifting deep bottleneck features for low-resource asr," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 5607–5611.

Phrase-based Language Modelling for Statistical Machine Translation

Achraf Ben Romdhane[1], *Salma Jamoussi*[1], *Abdelmajid Ben Hamadou*[1], *Kamel Smaïli*[2]

[1]MIRACL Laboratory, ISIM Sfax, Pôle Technologique, TUNISIA
[2]SMART team LORIA Nancy, FRANCE

achraf.ramdhan@gmail.com jamoussi@gmail.com
abdelmajid.benhamadou@isimsf.rnu.tn smaili@loria.fr

Abstract

In this paper, we present our submitted MT system for the IWSLT2014 Evaluation Campaign. We participated in the English-French translation task. In this article we focus on one of the most important component of SMT: the language model. The idea is to use a phrase-based language model. For that, sequences from the source and the target language models are retrieved and used to calculate a phrase n-gram language model. These phrases are used to rewrite the parallel corpus which is then used to calculate a new translation model.

1. Introduction

Machine translation systems have evolved since several decades from the use of a word to the use of a sequence of words (phrases) as basic units for translation. Currently, all the Statistical Machine Translation (SMT) systems are based on phrases. Succinctly, in the decoding step, the source sentence is segmented into phrases, each phrase is then translated into the target language and finally phrases are reordered [1]. At each step of the decoding phase, hypothesis are created and expanded until all words of the source sentence are covered. The expansion step produces a huge number of hypothesis which are constrained by the future cost estimation depending on the language model and the translation model probabilities. To achieve good translation quality, SMT researchers make a lot of effort in improving the translation model which moved from the original single-word-based models to phrase-based-models [1], in order to better capture the context dependencies of the words in the translation process. In the other hand and despite the improvements made in language modelling [2], [3], the state-of-the-art SMT systems use standard word n-gram models.

The idea, in this paper is to enhance the quality of SMT systems by improving their Language Models (LM). For that, we propose to use a phrase-based LM. This kind of models has already shown good performances in speech recognition tasks [4], [5] and we hope that it can help in the improvement of the machine translation task. In SMT, the language model is calculated on the target language. Then, to get a phrase-based language model, the target language model should be rewritten in terms of sequence of words. To do this, we propose to extract the source phrases using triggers [4]. We then use the inter-lingual triggers to retrieve the corresponding target sequences [6], [7]. Both source and target phrases are used to rewrite the parallel corpus which is used to train the language and the translation models. In section 2, we give an overview of the source phrase extraction method. Then in section 3, we present the method which associates to each source sequence its equivalent sequences in the target language. A description of the used corpora and the results achieved are presented and discussed in section 5 and 6. We end with a conclusion which points out the strength of our method and gives some tracks about future work in our research group.

2. Source phrases extraction

We use the concept of triggers [4],[7],[8] to extract pertinent sequences from a corpus. A trigger is composed of a word and its best correlated triggered words estimated in terms of mutual information (MI) :

$$I(x,y) = P(x,y)\log\frac{P(x,y)}{P(x)P(y)} \quad (1)$$

Where $P(x,y)$ is the joint probability and $P(x)$ and $P(y)$ are marginal probabilities. This allows to build a sequence of 2 words, to identify long phrases, an iterative process retrieves first, sequences of two words by grouping contiguous words which have a high MI then, in the second iteration, phrases of length 3 are identified, etc. To maintain a reasonable number of phrases, only the sequences which have a higher MI than the average MI of all sequences are kept for the forth coming steps. At the end of the process, we get a list of phrases which is used to rewrite the source corpus in terms of words and sequences. Examples of the retrieved phrases are given in table 1.

Since classical triggers allow to establish a triggering-triggered relationship between two events from the same language, Lavecchia et al. in [7] proposed to determine correlations between words coming from two different languages. These triggers called inter-lingual triggers. Each of them is

Phrases	**MI** $\times 10^{-5}$
parlement_européen	69.07
projet_européen	0.78
populaire_européen	0.22
politique_économique	0.17
commission_des_affaires_juridiques	0.039
commission_des_relations_économiquess	0.045
je_voudrais_vous_demander	0.032

Table 1: Examples of source phrases

composed of a triggering source event and its best correlated triggered target events.

3. Target phrases extraction

Once we have determined the list of the source sequences, we can then determine their corresponding sequences in the target side. For that, we used the method proposed by Lavecchia et al. [7] based on n-to-m inter-lingual trigger model. This method allows to associate to each source phrase of n words a set of target sequences of variable size m. In fact, for each source phrase of k words, we choose one or more target sequences of length k $\pm\Delta$k without performing any word alignment. In our case, for the language pair English-French, we set Δk to 1 in a way that a sequence of two words will be associated with the target sequences of length one, two or three words. Thus, we select for each source phrase the first 30 most relevant target sequences that have the best MI. An example of the extracted phrases with their best corresponding target sequences is presented in table 2.

Source phrases	Target phrases	**MI** $\times 10^{-2}$
parlement_européen	**european parliament**	**2.3**
	the european parliament	2.01
	parliament	1.7
	europen	1.6
	the european	1.3
je_vous_remercie	**thank you**	**0.43**
	thank you very much	0.091
	thank you for your	0.067
	i thank you	0.063
	very much	0.054

Table 2: Example of inter-lingual phrases

4. How to process the parallel corpus?

The objective in this section is to show how to rewrite both source and target copora in terms of phrases. For each source phrase, we select all possible target phrases by using inter-lingual triggers. The target phrases are added, in a decreasing order of MI, to a dictionary of phrases. Then the target corpus is rewritten in terms of these phrases. In case of conflict, the algorithm will prefer the phrase with the highest MI. At this point, we get a bilingual training corpus written in terms of word and phrases. The achieved corpora are then used to train the language and translation models. Table 3 illustrates some examples of sentences of the obtained training corpora.

thank_you_very_much for_your_attention . je_vous_remercie de_votre_attention .
thank_you_very_much for_your_contributions and support . merci de_vos_contributions et de votre soutien .
i declare the_session_of_the_european_parliament adjourned . je déclare interrompue la_session_du_parlement_européen .
adjournment of_the_session interruption de_la_session
a_new deal for_the_new world une_nouvelle donne pour le_nouveau monde
it_is easier in certain_areas . c'_est plus facile dans certains_domaines .

Table 3: Examples of sentences of the training corpora

5. Resources Used in IWSLT 2014

Training the translation and language models is constrained to data supplied by the organizers. For this campaign, we only participated in the English-French translation task.
Among the parallel data provided, we use WIT[3] [9] and EUROPARL [10]. As usual, we clean the raw data before performing any model training. This includes the lowercasing conversion and removing of long sentences. After the preprocessing operation, we get a parallel corpus of 1 767 644 sentences. The English side has a total of 35 million words (117006 unique tokens). The French side has a total of 38 millions words (141150 unique tokens).
A 5-gram language model has been trained with SRILM toolkit [11]. The word alignment of the parallel corpora is generated using GIZA++ Toolkit [12] in both directions. Afterwards, the alignments are combined using the grow-diag-final-and heuristic to obtain symetric word alignment model [1]. For decoding we used Moses toolkit [13] and the standard MERT to tune the weights of our features on the 100-best translation assumptions of the development set. Eight default features are used:

- Bidirectional phrase translation probability $(p(e|f), p(f|e))$
- Bidirectional lexical probability $(lex(e|f), lex(f|e))$
- Phrase penalty
- Word penalty
- Distortion model
- 5-gram language model

6. Experiments

6.1. The retrieved phrases

In this task, we set the maximum size of a phrase to 8 words, this is due to the fact that in previous experiments [14] phrases with more than 8 words do not contribute effectively in the improvement of the machine translation quality. The method described in 2 is applied in a way that at each iteration, we retrieve phrases of different lengths depending on the size S of the source phrase. To control that, we keep only target phrases of T words with $T = S \pm \Delta S$. For instance, in the first iteration, only sequences of T words (with $T \in \{1, 2, 3\}$) are kept.

We extracted from the French part of the training corpus, a set of 23064 phrases. Then, for each source phrase of S words, we kept the 30 best potential translations of size T. These sequences are included in the translation table and used to rewrite the training corpus. In this way, the target corpus is composed of single words and phrases of at maximum of 8 words. Consequently, training a 5-gram language model will take into account phrases up to 40 words (in the case of a 5-gram where each gram is composed of a phrase of 8 words).

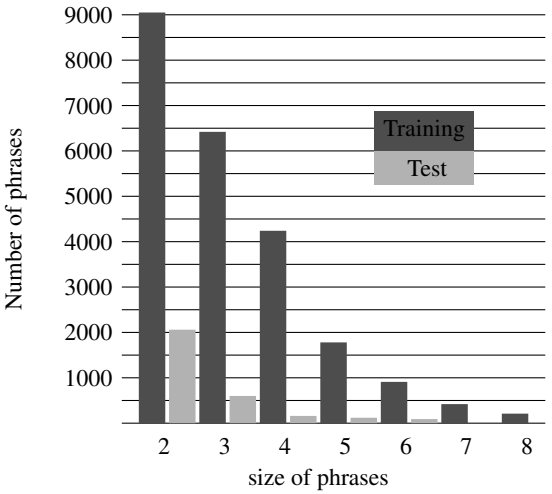


Figure 1: Histogram of the phrases number according to their size.

Figure 1 plots the histogram of the number of phrases contained in the training and the test corpora, according to their size. We can notice that the majority of phrases used are composed of two or three words which represents more than 60% of the extracted phrases. This histogram shows also that the number of phrases which occur in the test corpus is very low and does not exceed 12% of the whole extracted phrases.

6.2. Test data

The test data has to be written in the same way as the training corpus, for that two solutions are possible:

- Use the test corpora written in terms of words then we defragment our sequences belonging to the target part of the translation table.
- Rewrite the test data in terms of words and phrases. For this, the source sequences could be sorted according to their sizes or on their MI values. Then, for each sentence we explore the list of sequences in a decreasing manner. It worths mentioning that sorting sequences according to their size promotes the use of large size sequences while sorting sequence on their MI promotes the use of sequences short.

It should be noted that the system parameters were trained on the development corpus which combines the dev2010, tst2010 and tst2012. However we have chosen to report results on the tst2011, tst2013 and tst2014. Reported results are case-insensitive BLEU [15]. In addition, we performed tests on translation systems based on a training corpus written in terms of words and sequences:

- Sys1: uses a test corpus written in terms of words and sequences.
- Sys2: uses a test corpus written only in terms of words.

Table 4 illustrates the results obtained by different experiments on both development and test corpora.

System	Dev	tst11	tst13	tst14
baseline	28.91	36.84	-	-
Sys1	26.51	33.52	-	-
Sys2	28.27	35.48	30.91	26.97

Table 4: Results for the English → French MT task

On the development and the test corpus tst11, the use of a corpus written in terms of words (Sys2) is better than the one where the test data is rewritten in terms of phrases (Sys1). That's why, we decided to submit Sys2 as our primary SMT system. The small number of sequences used in our translation system and compared to the table of the baseline system is probably the reason which make our results worse than the baseline. Another explanation is related to the weak number of phrases contained in the test corpus, only 12% for tst13. Some translation examples are shown in Table 5.

7. Conclusions

In this paper, we evaluate our translation system on the data of IWSLT 2014 for English-French. Our contribution focuses on the use of a phrase-based language model and a translation model based on the phrases used in the language model. In order to train a phrase-based language model, we identified common source phrases by an iterative process.

Source	very often when i meet someone and they learn this about me there 's a certain kind of awkwardness .
Baseline	très souvent , lorsque je rencontre quelqu' un , et ils apprennent sur moi il y a une certaine gêne .
Sys2	très souvent quand je_rencontre quelqu' un , et ils_apprennent ce sur moi il y_a un_certain type_de gêne .
Reference	très souvent , quand je rencontre quelqu' un et qu' ils découvrent que je suis comme a , il y a un certain malaise .
Source	when we look at the population growth in terms of cars , it becomes even clearer .
Baseline	lorsque nous examinons la croissance de la population en termes de voitures , il devient encore plus clair .
Sys2	lorsque nous_examinons la_croissance_démographique en_termes de voitures , il devient encore plus clair .
Reference	quand nous regardons l' accroissement de la population en termes de voitures , ça devient même plus clair .

Table 5: Translation example from the tst11 set, comparing the baseline and the submitted system (Sys2) given a reference translation.

Then, we retrieved their potential translations by using interlingual triggers. These phrases are included in the translation table and used to rewrite the training corpus. The new corpus obtained is used to train the translation and language models. We evaluated the translation quality with the Bleu metric. The results showed that the state-of-the-art SMT system is better than our system. But, our results are encouraging and we plan to add some other features to the phrase based language model to improve the overall quality of our SMT system.

8. References

[1] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation", Proceedings of HLT-NAACL 2003, 2003, pp. 127-133.

[2] R. Sarikaya, Y. Deng, "Joint Morphological-Lexical Language Modeling for Machine Translation"', In Proceedings of NAACL HLT 2007, Companion Volume, 2007, pp. 145-148.

[3] M. Khalilov, "Improving target language modeling techniques for statistical machine translation", Proceedings of the Doctoral Consortium at the 8th EUROLAN Summer School, 2007, pp. 39-45.

[4] I. Zitouni, K. Smaïli, and J.-P. Haton, "Statistical language modeling based on variable-length sequences", Computer Speech and Language, vol. 17, 2003, pp. 27-41.

[5] S. Deligne, F. Bimbot, "Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams", In Proceedings ICASSP, 1995, pp. 169-172.

[6] W. Kim and S. Khudanpur, "Lexical triggers and latent semantic analysis for cross-lingual language model adaptation", ACM Transactions on Asian Language Information Processing (TALIP), 2004, pp. 94-112.

[7] C. Lavecchia, D. Langlois, K. Smaïli, "Discovering phrases in machine translation by simulated annealing", INTERSPEECH, ISCA, 2008, pp. 2354-2357.

[8] C. Tillmann, H. Ney, "Word Triggers and the EM Algorithm", In Proceedings of the Workshop Computational Natural Language Learning (CoNLL), 1997, pp. 117-124.

[9] M. Cettolo, C. Girardi and M. Federico, "WIT3: Web Inventory of Transcribed and Translated Talks", In Proceedings of EAMT,2012, pp. 261-268.

[10] P. Koehn, "Europarl: A multilingual corpus for evaluation of machine translation", In MT Summit, Thailand, 2005.

[11] A. Stolcke, "SRILM: An Extensible Language Modeling Toolkit," in Proceedings of the 7th International Conference on Spoken Language Processing, 2002, pp. 901-904.

[12] F. J. Och, and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models", Computational Linguistics, vol. 29, no. 1,2003, pp. 19-51.

[13] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statisti- cal machine translation," in Annual Meeting of the Association for Computational Linguistics (ACL) demonstration session, Prague Republic, 2007, pp. 177-180.

[14] C. Nasri, K. Smaïli, and C. latiri Training Phrase-Based SMT without Explicit Word Alignment, 15th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING), Nepal, 2014.

[15] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU:a method for automatic evaluation of machine translation", in Proceedings of ACL02, 2002, pp. 311-318.

The LIUM English-to-French Spoken Language Translation System and the Vecsys/LIUM Automatic Speech Recognition System for Italian Language for IWSLT 2014

Anthony Rousseau[1], *Loïc Barrault*[1], *Paul Deléglise*[1], *Yannick Estève*[1], *Holger Schwenk*[1]
Samir Bennacef[2], *Armando Muscariello*[2], *Stephan Vanni*[2]

LIUM[1], University of Le Mans, France
Vecsys[2], Montigny-le-Bretonneux, France
firstname.lastname@lium.univ-lemans.fr
{sbennacef, amuscariello, svanni}@vecsys.fr

Abstract

This paper describes the Spoken Language Translation system developed by the LIUM for the IWSLT 2014 evaluation campaign. We participated in two of the proposed tasks: (i) the Automatic Speech Recognition task (ASR) in two languages, Italian with the Vecsys company, and English alone, (ii) the English to French Spoken Language Translation task (SLT). We present the approaches and specificities found in our systems, as well as the results from the evaluation campaign.

1. Introduction

This paper describes the ASR and SLT systems developed by the LIUM for the IWSLT 2014 evaluation campaign. This year, the campaign has the particularity to bring new recognition languages and translation directions, while still proposing TED Talks recognition and translation tasks. Consequently, we participated in the two tasks mentioned above, with English and Italian languages for the ASR task; and English to French for the SLT task. Since we last participated in IWSLT three years ago in 2011, new approaches and specificities were developed by the LIUM, both in the ASR and in the SLT tasks, which will be detailed here. For ASR in Italian, this work was made in collaboration with the Vecsys company.

The remainder of this paper is structured as follows: in section 2, we describe the data used for both tasks and how the selection was performed. In section 3, we present the architecture of our ASR system and the results obtained on the various corpora used during the campaign. Then in section 4, we expose the architecture of our SLT system, along with its results during the campaign. Lastly, the section 6 concludes this system description paper.

2. Data Selection for the Tasks

Performance of Natural Language Processing (NLP) systems like the ones we are going to present here can often be enhanced using various methods, which can occur before, during or after the actual system processing. Among these, one of the most efficient pre-processing method is data selection, *i.e.* the fact to determine which data will be injected into the system we are going to build. For this campaign, many data selection processing was done, both in ASR and SLT tasks.

2.1. Selection for the ASR task

2.1.1. Acoustic models training data selection

For our acoustic modeling we used as a primary source the TED-LIUM corpus release 2 [1], removing from it all talks recorded after December 31st, 2010. In order to strengthen this base, we first added data from the Euronews corpora [2] distributed by the campaign organizers and from the 1997 English Broadcast News Speech (HUB4) [3]. Then, from the MediaEval 2014 evaluation campaign Search and Hyperlinking Task data transcripts (BBC recordings from 2008 which were decoded by the LIUM) [4], we applied a threshold on our confidence measures to select the best possible segments for our system within a limit of 50 hours of speech. Table 1 summarizes the characteristics of the data included in our ASR system acoustic models.

Corpus	Duration	Segments	Words
TED-LIUM	130.1h	61 796	1 447 022
Euronews	68.2h	33 686	817 649
1997 HUB4	75.0h	20 652	852 517
MedialEval 14	50.0h	46 713	368 118
Total	323.3h	162 847	3 485 306

Table 1: Characteristics of the acoustic data used in the LIUM ASR system acoustic models.

2.1.2. Language models training data selection

Since language models training data is constrained for the ASR task, we applied our data selection tool XenC [5] on each allowed corpus at our disposal: basically all of publicly available WMT14 data, a provided TED Talks closed-captions corpus and the LDC Gigaword. Based on cross-entropy difference from a corpus considered as in-domain and out-of-domain data, our tool allows to perform relevant data selection by scoring out-of-domain sentences regarding their closeness to the in-domain data. Table 2 summarizes the characteristics of the monolingual text data used to estimate our system language models.

Corpus	Original # of words	Selected # of words	% of Orig.
IWSLT14	0.1M	0.1M	100.00
Common Crawl	195.4M	13.6M	6.98
Europarl v7	56.4M	1.8M	3.22
Gigaword LDC	4 985.3M	168.2M	3.37
10^9 FR-EN	649.4M	11.9M	1.83
News Crawl	1 503.1M	44.8M	2.98
News-Comm. v7	4.7M	0.7M	14.04
UN 200x	360.1M	1.8M	0.50
Yandex 1M	24.1M	4.6M	19.01
Total (w/o IWSLT14)	7 778.5M	247.4M	3.18

Table 2: Characteristics of the monolingual text data used in the LIUM ASR system language models.

2.2. Data processing and selection for the SLT task

All available corpora have been used to train the different component of the SMT system. The source side of the bi-texts have been processed in order to make it more similar to speech transcriptions. After a standard tokenization, the processing mainly consisted in applying regular expressions to delete punctuations and unwanted characters, put capital letters in lowercase and rewrite numbers in letters.

Once the processing performed, monolingual and bilingual data selection has been applied using XenC [5]. For this purpose, the TED corpus has been used as in-domain corpus (to compute in-domain cross-entropy) and the provided development data (dev2010 and tst2010) was used to determine the quantity of data by perplexity minimization.

3. Automatic Speech Recognition Task in English

In this section, we will describe the Automatic Speech Recognition system developed by the LIUM for the IWSLT 2014 campaign, as well as present the results (both in-house and official) obtained on various corpora.

3.1. Architecture of the LIUM ASR system

Our system architecture is mainly based on the Kaldi open-source speech recognition toolkit [6] which uses finite state transducers (FSTs) for decoding. A first step is performed with the Kaldi decoder by using a bigram language model and standard GMM/HMM models to compute a fMLLR matrix transformation. A second decoding step is performed by using the same bigram language model and deep neural network acoustic models. This pass generates word-lattices: an in-house tool, derived from a rescoring tool included in the CMU Sphinx project, is used to rescore word-lattices with a 5-gram Continuous Space Language Model [7].

3.1.1. Speaker segmentation

To segment the audio recordings and to cluster speech segments by speaker, we used the *LIUM_SpkDiarization* speaker diarization toolkit [8]. This speaker diarization system is composed of an acoustic Bayesian Information Criterion (BIC)-based segmentation followed by a BIC-based hierarchical clustering. Each cluster represents a speaker and is modeled with a full covariance Gaussian. A Viterbi decoding re-segments the signal using GMMs with 8 diagonal components learned by EM-ML, for each cluster. Segmentation, clustering and decoding are performed with 12 MFCC+E, computed with a 10ms frame rate. Gender and bandwidth are detected before transcribing the signal. This speaker segmentation was used by all the LIUM and Vecsys ASR systems.

3.1.2. Acoustic modeling

The GMM-HMM (Gaussian Mixture Model - Hidden Markov Model) models are trained on 13-dimensions PLP features with first and second derivatives by frame. By concatenating the four previous frames and the four next frames, this corresponds to $39 * 9 = 351$ features projected to 40 dimensions with linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT). Speaker adaptive training (SAT) is performed using feature-space maximum likelihood linear regression (fMLLR) transforms. Using these features, the models are trained on the full 323.3 hours set, with 9 500 tied triphone states and 200 000 gaussians.

On top of these models, we train a deep neural network (DNN) based on the same fMLLR transforms as the GMM-HMM models and on state-level minimum Bayes risk (sMBR) as discriminative criterion. Again we use the full 323.3 hours set as the training material. The resulting network is composed of 7 layers for a total of 36.8 millions parameters and each of the 6 hidden layers has 2 048 neurons. The output dimension is 7 296 units and the input dimension is 440, which corresponds to an 11 frames window with 40 LDA parameters each. Weights for the network are initialized using 6 restricted Boltzmann machines (RBMs) stacked

as a deep belief network (DBN). The first RBM (Gaussian-Bernoulli) is trained with a learning rate of 0.01 and the 5 following RBMs (Bernoulli-Bernoulli) are trained with a rate of 0.4. The learning rate for the DNN training is 0.00001. The segments and frames are processed randomly during the network training with stochastic gradient descent (SGD) in order to minimize cross-entropy between the training data and network output. When these training steps are done, the last step of training is processed, by applying the minimum Bayes risk criterion, as indicated above. To speed up the learning process, we use a general-purpose graphics processing unit (GPGPU) and the CUDA toolkit for computations.

3.1.3. Language modeling

For language modeling, we rely on the SRILM language modeling toolkit [9] as well as the Continuous Space Language Model toolkit. The vocabulary used in the LIUM ASR system is composed of 165 371 entries. The bigram language model (2G LM) used during the Kaldi decoding part is trained on the data presented in section 2.1.2.

With the SRILM toolkit, one 2G LM is estimated for each corpus source, with no cut-offs and the modified Kneser-Ney discounting method. These 2G LM are then linearly interpolated to produce the final 2G LM which will be used in the final system, using the IWSLT 2011 development and test corpora. To rescore the word-lattices produced by Kaldi, a trigram and a quadrigram language models (3G and 4G LM) are estimated with the same toolkit, again by training one LM by corpus source and then linearly interpolating them. A 5G continuous-space language model (CSLM) is also estimated for the final lattice rescoring, with no cut-offs and the same discounting method as for the bigram language model. The table 3 details the interpolation coefficients for the 2G, 3G and 4G language models as well as the final perplexity for each final model.

Corpus	Coefficients			
	2G	3G	4G	CSLM
IWSLT14	.36353	.23963	.19110	N/A
Common Crawl	.14404	.26584	.34979	N/A
Europarl v7	.00272	.00244	.00277	N/A
Gigaword LDC	.30076	.27450	.24411	N/A
10^9 FR-EN	.02709	.02882	.02701	N/A
News Crawl	.13535	.14751	.14241	N/A
News-Comm. v7	.00173	.00254	.00220	N/A
UN 200x	.00300	.00411	.00391	N/A
Yandex 1M	.02179	.03461	.03670	N/A
Perplexity	209.31	134.38	107.72	123.03

Table 3: Interpolation coefficients and perplexities for the bigram, trigram, quadrigram and CSLM language models used in the LIUM ASR system.

3.2. Results

We submitted three runs (one primary, two constrastives) for the ASR task. The first contrastive is the one described in section 3.1. The second constrastive is basically the same system as the first, with a DNN similar the the CRIM one described in [10]. The primary is the fusion of the two systems described above at the word-lattices level. The table 4 presents the official results from the campaign organizers. Rankings are not known at the time of this paper publication.

System	tst2014	tst2013
Primary	12.3 %	16.0 %
Contrastive 1	13.4 %	17.3 %
Contrastive 2	13.8 %	17.4 %

Table 4: Official results (Word Error Rate) for the LIUM at the IWSLT 2014 Automatic Speech Recognition track.

4. Spoken Language Translation Task

In this section, the architecture of our Statistical Machine Translation (SMT) system used in the SLT task is described.

4.1. Architecture of the LIUM SLT system

The SMT system is based on the Moses toolkit [11]. The standard 14 feature functions were used (*i.e* phrase and lexical translation probabilities in both directions, seven features for the lexicalized distortion model, word and phrase penalty and target language model (LM) probability). In addition to these, an Operation Sequence Model (OSM) [12] have been trained and included in the system.

4.1.1. Translation model

The translation models have been trained with the standard procedure. First, the bitexts are word aligned in both directions with GIZA++ [13]. Then the phrase pairs are extracted and the lexical and phrase probabilities are computed. The weights have been optimized with MERT using two versions of the development data. For some systems, the provided transcriptions were used, and for others, the outputs of our ASR system was used. This was performed for the sake of comparing the impact of ASR systems improvement (observed during the last few years).

4.1.2. Language modeling

The language model is an interpolated 4-gram back-off LM trained with SRILM [9] on the selected part of the French corpora made available. The vocabulary contains all the words from the development sets, the target side of bitexts and only the more frequent words from the large monolingual corpora. The interpolation coefficient have been optimized using the standard EM procedure. The perplexity of

this model was 69.37.

In addition, several large context CSLM [14] have been trained, each with a different architecture. Those models are used (alone or in combination) to rescore the n-best list of SMT hypotheses. The weights for the CSLM have been optimized with CONDOR [15], a numerical optimizer, with $-$BLEU as objective function to minimize.

Name	Order	Projection Layer	Perplexity
CSLM1	12	384	40.72
CSLM2	12	448	40.19
CSLM3	16	384	40.58

Table 5: Architecture of the various CSLM trained for rescoring the n-best list of SMT hypotheses.

4.1.3. Submitted systems

A total of six systems were submitted for evaluation. One of the differences lies in the development data used for tuning. The provided development data corresponds to ROVER outputs of several years old ASR systems. Considering that ASR systems have greatly evolved during the last few years, we thought that comparing an SMT system tuned with outputs of an old combination of ASR systems with a state-of-the-art ASR system would be interesting. The other difference concerned the use of a rescoring step. As mentioned in the previous section, several CSLM have been trained. Some systems did not include any rescoring process at all, some use only one CSLM and some combined the three CSLM probabilities to determine the best hypothesis. When using only one CSLM, the best performing model on the development data has been chosen. The results and discussion are presented in the next section.

4.2. Results and discussion

The results obtained on the development and test data are presented in Table 6.

We translated two version of the test data. $test2014 - iwslt$ is the provided test data, which corresponds to a ROVER combination of the outputs of the systems participating in the IWSLT'14 ASR task. $test2014 - lium$ corresponds to the 1-best output of the LIUM ASR system.

The first comment is that the results that we observed on the development data are not reflected in the test data. Tuning with two versions of the development data, providing difference of more than 2 BLEU points results in similar scores on the test data. This is well understood when there is a mismatch between tuning and testing conditions (i.e. tuning with $-lium$ corpus and testing on $-iwslt$). As the ASR results have not been provided yet, we can't make the link between the WER and the SMT results. Also, a deeper analysis of the outputs have to be performed in order to explain this behavior.

The main improvements are obtained by rescoring the 1000-best list of hypotheses with one or more CSLM. By comparing Contrast2 and Contrast4 systems on one hand, and Contrast4 and Contrast6 systems on the other hand, we can observe that CSLM rescoring provide a gain of up to 1.2 and 1.78 BLEU respectively on the development and test data.

However, combining the three different CSLM does not provide anymore gain. This was already observed on the development data, but the result was never worse than using only one language model. This tends to prove that CSLM with different architectures (context and projection layer size in this case) does not have a great impact on the final score.

5. Automatic Speech Recognition Task in Italian

The ASR system used to process Italian data is a combination of the Vecsys ASR system and the LIUM ASR system. Both systems share the same speaker segmentation and the same training data, very restricted in the ASR task for Italian. The speaker diarization system is the same as the one used to process English data.

5.0.1. Training data

To train language models for Italian, the number of authorized sources of training data was very low. We used the data provided by the organizers to train language and acoustic models, in addition to the Italian Google n-grams, listed in the permissive data (LDC2009T25). For acoustic models, in addition to the Euronews corpora [2] distributed by the organizers, we used about 100 hours of manually annotated data owned by the Vecsys company, and recorded before June 30th 2011. Notice that we extracted about 75h from the Euronews automatically annoted data: about 175 hour of recordings were used to train the acoustic models of the Vecsys and LIUM systems.

5.1. Vecsys system

Vecsys speech recognition system is based on a multi-pass GMM/HMM decoding of the input speech, mostly derived from the CMU Sphinx toolkit. A first pass aims at providing a transcription which, in accordance with the speaker segmentation, is employed to estimate speaker-specific fMLLR matrices for feature transformation. The transformed features are used in a second decoding that produces word lattices, using the same trigram back-off language model as for the first pass, and then acoustically rescored to improve inter-word senone scores. The final transcription is obtained by joint linguistic rescoring of the word lattices from a 4 gram back-off and a 4-gram continuous space language model, followed by a confusion network decoding.

Name	CSLM	Dev		test2014-iwslt				test2014-lium			
				Case		No-Case		Case		No-Case	
		Name	%BLEU	%BLEU	%TER	%BLEU	%TER	%BLEU	%TER	%BLEU	%TER
Primary	Comb	d10t10lium	25.79	26.82	59.40	27.85	57.69	24.90	60.93	25.92	59.55
Contrast1	Comb	d10t10iwslt	23.30	26.78	59.40	27.82	57.99	25.06	61.10	26.04	59.73
Contrast2	CSLM1	d10t10lium	25.70	26.76	58.82	27.81	57.46	24.98	60.72	25.99	59.33
Contrast3	CSLM3	d10t10iwslt	23.24	26.89	59.36	27.94	57.94	24.95	61.43	25.96	59.97
Contrast4	-	d10t10lium	24.49	25.17	59.83	26.17	58.47	23.59	61.66	24.52	60.26
Contrast5	-	d10t10iwslt	22.26	25.14	60.61	26.16	59.21	23.65	62.24	24.64	60.82

Table 6: Results obtained with the submitted systems. Corpora d10t10lium and d10t10iwslt correspond to respectively the transcription obtained with the LIUM ASR system and the provided development data.

5.1.1. Acoustic modeling

Italian phonetic lexicon is describes by a set of 27 phonemes, and, for all consonants, gemination is modeled by doubling the consonant symbol in a word pronunciation, rather than defining a special symbol for the geminate consonant. The GMM-HMM acoustic models are computed from 13-dimensional PLP features (including energy) to which first and second order derivates are appended. In all decoding steps, PLPs are multiplied by an LDA and a MLLT matrix, both estimated on the same training data, to obtain a 29-dimensional vector. The first pass uses a light-weight set of models which comprises 6000 senones, each modeled by a 16-component GMM, estimated by ML modeling, and adapted by MAP according to gender. The second pass uses 8000 senones, each modeled by a 32-component GMM, estimated by MPE modeling from an initial set of m/f SAT models.

5.1.2. Language modeling

Back-off language models are obtained by interpolation of two back-off models, one estimated by Witten-Bell discounting on the Google N-gram corpus, the other from Kneser-Ney discounting and no cut-off on the TED transcriptions provided by the organizers. On this same corpus, a 4 gram continuous space language model is trained: scores are computed for 4-grams of words included in a short list of 16384 words out of 109500 words. Such scores are linearly interpolated with those read from the back-off model for the corresponding 4 grams.

5.2. LIUM ASR system for Italian

The architecture of the LIUM ASR system for Italian is the same as the one described in this paper for English language. The phonetic lexicon was built from the lexicon provided in the Festival tool for speech synthesis [16], by using the statistical grapheme-to-phoneme (g2p) tool described in [17] in order to compute the pronunciation of words not included in the Festival Italian lexicon. This Festival lexicon contains about 400,000 words.

5.3. Merging Vecsys and LIUM ASR systems

Vecsys and LIUM used the same audio segmentation, provided by the *LIUM_SpkDiarization* speaker diarization system. Using the same segmentation makes easier the merging between the two ASR outputs: final outputs were obtained by merging word-lattices provided by both ASR systems, as described in [18].

6. Conclusion

We presented the LIUM's and Vecsys' ASR and SMT systems which participated in the ASR and SLT tracks of the IWSLT'14 evaluation campaign.

By integrating some of the latest LIUM developments in Automatic Speech Recognition, we were able to achieve a Word Error Rate score of 12.3 % on the ASR evaluation track. While we currently can't compare it to other results for the tst2014 corpus, we can compare the 16.0 % tst2013 score to the last year results, which would have been ranked 4th.

By rescoring with a continuous space language model, we obtained a gain of about 1.7% BLEU on the SLT test data. However, the combination of several CSLM rescoring did not produced anymore gain.

7. Acknowledgements

This work was partially funded by the European Commission through the EUMSSI project, under the contract number 611057, in the framework of the FP7-ICT-2013-10 call. This work was also partially funded by the French National Research Agency (ANR) through the TRIAGE project, under the contract number ANR-12-SECU-0008-01.

8. References

[1] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014.

[2] R. Gretter, "Euronews: a multilingual speech corpus for ASR," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014.

[3] J. Fiscus, J. Garofolo, M. Przybocki, W. Fisher, and D. Pallett, "1997 English broadcast news speech (HUB4) LDC98S71," Web Download, 1998.

[4] M. Eskevich, R. Aly, D. N. Racca, R. Ordelman, S. Chen, and G. J. Jones, "The search and hyperlinking task at MediaEval 2014," in *Working Notes Proceedings of the MediaEval 2014 Workshop*, Barcelona, Spain, october 2014.

[5] A. Rousseau, "XenC: An open-source tool for data selection in natural language processing," *The Prague Bulletin of Mathematical Linguistics*, vol. 100, pp. 73–82, 2013.

[6] D. Povey, A. Ghoshal, G. Boulianne, L. Burge, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, december 2011, iEEE Catalog No.: CFP11SRW-USB.

[7] H. Schwenk, "CSLM - a modular open-source continuous space language modeling toolkit," in *Interspeech*, august 2013, pp. 1198–1202.

[8] S. Meignier and T. Merlin, "LIUM SpkDiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, Dallas, Texas, USA, 2010.

[9] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of Interspeech*, September 2002, pp. 901–904.

[10] A. Rousseau, G. Boulianne, P. Deléglise, Y. Estève, V. Gupta, and S. Meignier, "LIUM and CRIM ASR system combination for the REPERE evaluation campaign," in *Text, Speech and Dialogue*. Springer International Publishing, 2014, pp. 441–448.

[11] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics (Demo and Poster Sessions)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 177–180. [Online]. Available: http://www.aclweb.org/anthology/P/P07/P07-2045

[12] N. Durrani, H. Schmid, and A. Fraser, "A joint sequence translation model with integrated reordering," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 1045–1054.

[13] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, ser. SETQA-NLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 49–57. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622110.1622119

[14] H. Schwenk, "Continuous space language models," *Computer Speech and Language*, vol. 21, pp. 492–518, 2007.

[15] F. Vanden Berghen and H. Bersini, "Condor, a new parallel, constrained extension of powell's uobyqa algorithm: Experimental results and comparison with the dfo algorithm," *J. Comput. Appl. Math.*, vol. 181, no. 1, pp. 157–175, Sept. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.cam.2004.11.029

[16] P. Cosi, F. Tesser, R. Gretter, C. Avesani, and M. Macon, "Festival speaks Italian!" in *INTERSPEECH*, 2001, pp. 509–512.

[17] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[18] A. Rousseau, G. Boulianne, P. Deléglise, Y. Estève, V. Gupta, and S. Meignier, "LIUM and CRIM ASR System Combination for the REPERE Evaluation Campaign," in *Text, Speech and Dialogue*. Springer, 2014, pp. 441–448.

LIMSI English-French Speech Translation System

Natalia Segal[1], *Hélène Bonneau-Maynard*[1,2], *Quoc Khanh Do*[1,2],
Alexandre Allauzen[1,2], *Jean-Luc Gauvain*[1], *Lori Lamel*[1], *François Yvon*[1]

LIMSI-CNRS[1] and University Paris-Sud[2], rue John von Neumann, F 91403 Orsay

firstname.lastname@limsi.fr

Abstract

This paper documents the systems developed by LIMSI for the IWSLT 2014 speech translation task (English→French). The main objective of this participation was twofold: adapting different components of the ASR baseline system to the peculiarities of TED talks and improving the machine translation quality on the automatic speech recognition output data. For the latter task, various techniques have been considered: punctuation and number normalization, adaptation to ASR errors, as well as the use of structured output layer neural network models for speech data.

1. Introduction

LIMSI participated in the IWSLT 2014 Evaluation Campaign in the spoken language translation (SLT) task for English→French language pair. Although LIMSI hosts both automatic speech recognition (ASR) and machine translation (MT) research activities, this was our first contribution to the SLT task and the effort was thus focused on one single translation direction. This year's SLT task consists in automatic transcription and translation of a test set composed of several recordings of TED online conferences[1]. The automatic speech transcriptions that have been used in our experiments were produced by the in-house ASR system adapted to TED data, rather than using the transcripts provided by the organizers (hypotheses from several automatic speech recognizers combined using the ROVER approach). As far as the automatic translation step is concerned, we addressed various typical challenges of SLT: to bring automatic transcriptions closer to the expectations of the MT system (mainly trained on written text), to adapt MT models to erroneous ASR output, and to improve the general translation quality.

This paper is structured as follows. We first present the ASR system and the adaptation steps taken to improve the automatic transcriptions of the TED data. We then describe various approaches used to bring the ASR output data and the expected MT input data format into accordance with each other, as well as our attempts to adapt standard MT systems to ASR output. Finally, the impact of re-scoring n-best translation hypotheses using SOUL models is presented in the closing section.

[1] https://www.ted.com/

2. ASR systems: adaptation to TED talks data

The LIMSI automatic speech recognition system for broadcast data [1] was adapted to the task of transcribing TED talks. The adaptations concern the acoustic and language models and the pronunciation dictionary.

Prior to transcription, the audio documents are partitioned identifying the portions containing speech to be transcribed [2] and associating segment cluster labels, where each segment cluster ideally represents one speaker.

Two types of acoustic features are used. The first are PLP-like [3], with cepstral normalization carried out on a segment-cluster basis [1]. A 3-dimensional pitch feature vector (pitch, Δ and $\Delta\Delta$ pitch) is added to the original PLP one, resulting in a 42-dimension feature vector. The second type are probabilistic features produced by a Multi-Layer Perceptron (MLP) from raw TRAP-DCT features [4], which have been shown to improve system performance when concatenated with cepstral features [5]. The MLP networks were trained using the simplified training scheme proposed in [6] using phone state targets. The feature vector formed by concatenating the MLP, PLP and pitch features has 81 elements.

The acoustic models are gender-independent, tied-state, left-to-right 3-state HMMs with Gaussian mixture observation densities (typically 32 components). The triphone-based phone models are word-independent, but position-dependent. The states are tied by means of a decision tree to reduce model size and increase triphone coverage. The acoustic models are speaker-adaptive (SAT) and Maximum Mutual Information (MMIE) trained.

N-gram language models are obtained by interpolating multiple unpruned component LMs trained on subsets of the training texts and used for both decoding and lattice rescoring. Language model training is performed with LIMSI STK toolkit which allows efficient handling of huge language models without any pruning or cutoff.

Word decoding is carried out in two passes. Each decoding pass produces a word lattice with cross-word, word-position dependent acoustic models, followed by consensus decoding with a 4-gram language model and pronunciation probabilities. The system vocabulary contains 95k words. Unsupervised acoustic model adaptation is performed for each segment cluster using the CMLLR and MLLR [7], and the lattices produced are rescored with a 4-gram back-off

dataset	WER (del., ins.)
dev2010	15.0 (4.0, 3.5)
tst2010	12.7 (3.3, 2.7)

Table 1: Case-insensitive recognition results on the 2010 dev and tst data, scored using sclite.

LM. The first decoding pass is carried out with a modified version of our 2011 Quaero system for broadcast data in English [8, 9] in which a language model trained on the provided ASR texts including the IWSLT14 TED LM transcriptions (3.2M words) was interpolated with the baseline 78k-word language model. The first decoding pass is done in 1xRT. The acoustic models in the first pass were trained on the data distributed in Quaero as well as on data from other sources from previous European or national projects and from the LDC. All acoustic and other language model training data predate December 31, 2010. The Euronews data provided by the organizers was not used. The second pass decoding used the same interpolated language model with acoustic models trained only on 180 hours of transcribed TED talks predating December 31, 2010 to better target the TED data.

The case-insensitive recognition results on the 2010 dev and tst data are given in Table 1 scoring with the NIST sclite scoring using the provided stm and no glm.

3. MT systems: adaptation to speech data

3.1. Machine Translation with N-code

NCODE implements the bilingual n-gram approach to SMT [10, 11, 12] that is closely related to the standard phrase-based approach [13]. In this framework, the translation is divided into two steps. To translate a source sentence $\mathbf{f}$ into a target sentence $\mathbf{e}$, the source sentence is first reordered according to a set of rewriting rules so as to reproduce the target word order. This generates a word lattice containing the most promising source permutations, which is then translated. Since the translation step is monotonic, the peculiarity of this approach is to rely on the n-gram assumption to decompose the joint probability of a sentence pair in a sequence of *bilingual* units called *tuples*.

The best translation is selected by maximizing a linear combination of feature functions using the following inference rule:

$$\mathbf{e}^* = \underset{\mathbf{e},\mathbf{a}}{\operatorname{argmax}} \sum_{k=1}^{K} \lambda_k f_k(\mathbf{f}, \mathbf{e}, \mathbf{a}), \qquad (1)$$

where K feature functions (f_k) are weighted by a set of coefficients (λ_k) and where $\mathbf{a}$ denotes the set of hidden variables corresponding to the reordering and segmentation of the source sentence. Along with the n-gram translation models and target n-gram language models, 13 conventional features are combined: 4 *lexicon models* similar to the ones used in standard phrase-based systems; 6 *lexicalized reordering models* [14, 15] aimed at predicting the orientation of the next translation unit; a "weak" distance-based *distortion model*; and finally a *word-bonus model* and a *tuple-bonus model* which compensate for the system preference for short translations. Features are estimated during the training phase. Training source sentences are first reordered so as to match the target word order by unfolding the word alignments [12]. Tuples are then extracted in such a way that a unique segmentation of the bilingual corpus is achieved [11] and n-gram translation models are then estimated over the training corpus composed of tuple sequences made of surface forms or POS tags. Reordering rules are automatically learned during the unfolding procedure and are built using part-of-speech (POS), rather than surface word forms, to increase their generalization power [12].

3.2. MT baseline

This section describes the MT systems trained on written material that served as a benchmark for the succeeding experiments aiming at improving the translation quality for speech transcriptions.

All the parallel corpora used in our translation systems have been preprocessed to remove excessively long sentences as well as sentences with an important length difference between the source and the target. The common preprocessing also included tokenization using the in-house tool described in [16] and word alignments using MGIZA++ [17] and Moses's grow-diag-final-and heuristic for alignment symmetrization.

All the MT systems developed in this study make use of the N-code system described above for translation model training and for decoding. Since the N-code system uses factored models, the training corpora have been tagged with part-of-speech (POS) labels using TreeTagger [18]. The target language model used discriminative log-linear interpolation approach to combine the model trained on TED monolingual data provided by the organizers and the bigger LM trained on WMT data (SRILM [19] toolkit was used for both models).

Our baseline system only uses the training data provided by the IWSLT campaign organizers, composed exclusively of TED talks recordings: we were thus subsequently able to quickly experiment with various adaptation techniques as well as to measure the impact of including large, out-of-domain, corpora.

We performed some additional cleaning on TED corpus, mostly related to extra textual information not present in the audio signal: removing speaker names or initials at the beginning of some lines, removing comments between square brackets and between parentheses, etc. Those notes are added by transcribers in order to facilitate the understanding of the text by human readers, but are useless and even confusing in the context of automatic speech translation.

3.2.1. Impact of the out-of-domain corpora

We tried to improve the performance of the baseline system trained on in-domain data only, by adding various bilingual corpora from the WMT Evaluation Campaign [20]: News-Commentary (NC), Europarl (EPPS) and Gigaword filtered as in [21] (GIGA). All those models were tuned on the same manually transcribed development set (dev2010). As can be seen in Table 2, only the filtered Gigaword corpus actually helped improve the performance of the baseline system. In accordance with these results, we used only this corpus as the additional out-of-domain corpus for our final system.

Table 2: *Baseline MT experiments with written corpora.*

training corpora	BLEU	
	dev2010	test2010
TED	28.8	33.2
TED + NC + EPPS	29.5	33.0
TED + NC + EPPS + GIGA	29.6	34.0
TED + GIGA	29.7	**34.4**

For the sake of speeding up the experiments with the adaptation of the MT systems to the characteristics of the speech data, only the TED corpus was used for training those intermediate systems. Our final system, however, to which the SOUL re-scoring was applied, made use of both TED and the Gigaword data.

3.3. Narrowing the gap between ASR and MT

An important source of MT quality deterioration on ASR output consists in various formatting differences between this output and the written corpora used for the training of the MT engine. One of the promising axes of improving the speech translation quality is therefore to reduce the gap between the ASR output and the source part of the parallel corpora. This goal can be achieved both by post-processing the speech recognition output before translation and by modifying the source part of the corpora used in MT training to make them more alike. In this work, we have experimented with two types of such processing: normalization of numbers and punctuation insertion. Other types of normalization might of cause be considered, such as the normalization of units of measurement, dates, acronyms etc.

3.3.1. Normalization of numbers

One inconsistency between the output of the ASR system and the expected input of the MT system is the fact that the speech recognition system produces the numbers spelled out, whereas MT systems are trained on written texts where numbers are usually written in digits. In both cases, the choice of the approach to number processing is optimal for the corresponding system: a fully spelled representation is closest to the pronunciation (big numbers may correspond to several pronounced words) and is thus convenient for ASR; digital representation is best suited for MT since it is much easier to translate to the equivalent digital representation on the target side. For speech translation, however, the inconsistency in number representations is one obvious source of the translation quality's deterioration. To transform fully spelled numbers in the ASR output into digits, we used a rule-based algorithm provided by LIMSI's ASR system as part of the post-processing to the main recognition system. It must be noted, however, that the numbers in written texts and the numbers produced via the above processing are not always the same. On the one hand, the automatically produced digital forms may contain errors, and on the other hand, human transcriptions are not always consistent and can choose either to spell out or not some of the numbers (*e.g. 1/3rd vs. one-third*). To bring ASR output as close as possible to the expectations of MT, we applied the number transformation to the source side of the TED corpus. In order to do this, we first converted all the digital numbers to text and then re-converted them to digits using the same algorithm as for the post-processed ASR output. A new MT system was then trained based on this corpus (norm).

To evaluate the impact of the number normalization on speech translation, we used the test set provided by the organizers (tst2010), for which we compared the translation performance on manual transcriptions to the performance on the automatic transcriptions produced by our baseline ASR system (WER=17%). Table 3 compares the performance of the baseline system to the performance of the system trained and tuned on normalized corpora. As expected, on the ASR output better results were obtained with normalization. However, the results on the manual transcriptions suffered a small degradation which is most probably due to to the errors produced by the normalization processing.

Table 3: *Experiments with number normalization.*

training corpora	normalization	BLEU (tst2010)	
		auto	manual
TED	no norm	20.5	33.2
	norm	**21.0**	33.0

3.3.2. Punctuation

Speech speech recognition systems do not generally produce punctuation as part of their output. The LIMSI ASR system makes it possible to add punctuation in a post-processing step, but it only includes very basic punctuation marks, such as commas and stop signs. The MT system, on the other hand, is expected to produce fully punctuated text as its output and is typically trained on punctuated sources. The performance on the manually transcribed test data, that does not contain any recognition errors, is nevertheless degraded dramatically if the punctuation is removed from the source side of the test (BLEU=25.5, as compared to BLEU=33.0 for the punctuated test, see Table 3).

Possible solutions to this problem have been explored, for example, in [22]. One solution is to build a new MT system based on the training corpora with unpunctuated source side: the system is thus trained to implicitly insert punctuation as part of the general translation process (implicit punctuation). Another solution is to produce automatic punctuation for the source language and to insert some punctuation marks to speech recognition output before translation (explicit punctuation in source): this approach has the advantage of allowing to keep the MT system unchanged. Our experiments with both approaches are shown in Table 4. We trained a new MT system unpunctuated in source (implicit punct), where we removed all the punctuation marks from the source side of both training corpus (TED) and tuning corpus (dev2010). This unpunctuated system was applied to the normalized ASR output without punctuation in test. The punctuated version of the TED MT system was applied to the same test punctuated by one of our two punctuation systems. Both of these punctuation systems were based on MT techniques and were trained on unpunctuated TED corpus as source and the same corpus with punctuation in target. One system used all the possible punctuations (all), whereas the other only used simple unpaired punctuation: commas, stops, colons, semi-colons, question and exclamation marks (main). The implicit punctuation and as well as the explicit punctuation with main marks achieve equivalent performance on test corpus. The fact that main punctuation insertion yields in better performance than all punctuation insertion can be explained by the fact that the paired punctuation marks (such as quotes or parentheses) are often separated by several words and are therefore much harder to predict correctly in the MT framework. The data sparsity also contributes to the fact that the insertion of all the types of the punctuation may add more errors than correct predictions.

Table 4: *Experiments with punctuation.*

training corpora	punct test	BLEU (tst2010 auto)
TED (implicit punct)	none	**24.4**
TED (man punct)	none	21.0
	auto all	24.0
	auto main	**24.4**

3.4. Adaptation of MT systems to ASR output

In addition to various surface differences between ASR output and MT training corpora such as described above, the most important source of difficulties for speech translation are the errors and the irregularities present in speech recognition output: if the source is degraded, the quality of translation is likely to suffer subsequently. It is to be expected, however, that for some types of errors the translation quality could be improved if the training data for MT included the errors produced by the recognizer, thus allowing for the MT system adapt to the variation in the output of this specific recognizer. This is why we experimented with an extra training corpus (TED auto) obtained by automatic transcription of the speech signal of the talks present in TED training corpus by our baseline ASR system. The corpus thus produced was normalized as described above. Since both punctuated and unpunctuated versions of the manual TED training corpus produced similar results and for the sake of time, we used only the unpunctuated version for these experiments so as to quickly determine the impact of the ASR output in training.

Table 5 compares different configurations for training corpora:

- TED manual transcription only
- TED auto transcription only
- TED manual and TED auto used separately (two translation tables)
- TED manual and TED auto used together (one translation table)

The source side of the development corpus (dev2010) was composed of manual transcriptions for the first model, of automatic transcriptions for the second model and of both automatic and manual transcriptions for the last two models.

Using both corpora produces the best results probably since it allows for the MT system to learn on both correct and erroneous examples. The best performance is achieved with one translation table.

Table 5: *Adaptation to ASR output in MT training.*

training corpora	BLEU (test2010 auto, no punct)
TED man only	24.4
TED auto only	24.2
TED man+auto (2 tables)	24.6
TED man+auto (1 table)	**24.8**

3.5. Final MT system configuration

Based on the results of all the experiments with speech translation described above, for the final systems we used two corpora in training:

- TED man+auto (in one corpus)
- Gigaword (filtered)

Table 6 presents the results for these systems both with and without punctuation in source. The performance of the punctuated system (with ASR data re-punctuated by *punct main*) proved to be slightly better, so this system was used for the final step of the processing: SOUL NNLM and NNTM n-best re-scoring. This table also reports the performance of the final punctuated MT system on the test set transcribed with the final ASR system adapted to TED data (WER=12.8%),

as compared to the same test set transcribed with the baseline ASR system (WER=17%). This shows the impact of the ASR quality on the translation performance. We subsequently used this test set for the experiments with SOUL.

Table 6: *Final MT system performance and the impact of the ASR adaptation to TED data on the MT performance.*

training corpora	punctuation	BLEU (test2010 auto)	
		ASR baseline	ASR final run
TED man+auto (1 table)	no punct	24.8	-
+ GIGA	no punct	25.0	-
	punct main	25.5	**27.7**

3.6. SOUL models

Neural networks, working on top of conventional n-gram back-off language models, have been introduced in [23, 24] as a potential means to improve discrete language models. As in previous submissions in the WMT evaluation (see [25] for instance), we took advantage of the recent proposal of [26]. Using a specific neural network architecture, the *Structured OUtput Layer* (SOUL), it becomes possible to estimate n-gram models that use large vocabulary, thereby making the training of large neural network models feasible both for target language models and for translation models [27]. Moreover, the peculiar parameterization of continuous models allows us to consider longer dependencies than the one used by conventional n-gram models, for instance $n = 10$ instead of $n = 4$.

3.6.1. Description of model structure

SOUL language model is a feed-forward multilayer neural networks estimating word's probability given its context made of the $n-1$ previous words (typically $n = 10$). While this model is similar to neural probabilistic language models introduced in [23], the output layer that predicts the word is organized into a tree structure. This structured output layer allows the model to predict words for large vocabulary applications.

SOUL translation models rely on a specific decomposition of the joint probability $P(\mathbf{f}, \mathbf{e}, \mathbf{a})$ of a sentence pair, where $\mathbf{f}$ is a sequence of I *reordered* source words $(f_1, ..., f_I)$, and $\mathbf{e}$ contains J target words $(e_1, ..., e_J)$, and $\mathbf{a}$ is an alignment between $\mathbf{f}$ and $\mathbf{e}$. In the n-gram approach to SMT [10, 11, 12] this segmentation is a by-product of source reordering, and ultimately derives from initial words and phrase alignments. In this framework, the basic translation units are tuples, which are analogous to phrase pairs, and represent a matching $u = (\overline{f}, \overline{e})$ between a source phrase $\overline{f}$ and a target phrase $\overline{e}$.

The n-gram assumption decomposes the joint probability into the products of tuples' probabilities as follow:

$$P(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \prod_{i=1}^{L} P(u_i | u_{i-1}, ..., u_{i-n+1}) \quad (2)$$

However, as mentioned in [27], this decomposition implies a large vocabulary of bilingual tuples, hence its generalisation capability is limited due to data sparsity issues. As a remedy, the n-gram probabilities in the right-hand side of (2) are factored by first decomposing tuples into source and target parts (or phrases), and then considering each part as a word stream. The decomposition process results in 4 word-factored bilingual models as described in [27], each of which produces a feature score that is added to the final system before SOUL (Section 3.4).

3.6.2. Integration of SOUL *models*

Given the computational cost of computing n-gram probabilities with neural network models, we resorted to a two-pass approach: the first pass uses a conventional system to produce an N-best list (the N most likely hypotheses); in the second pass, probabilities are computed by SOUL models for each hypothesis and added as new features. Then the N-best list is reordered according to a combination of all features including these new features. In our experiments, 10-gram SOUL models were used to rescore 300-best lists. Overall system's log-linear coefficients were optimised using k-best Batch Margin Infused Relaxed Algorithm (KBMIRA) [28] on the automatically transcribed development set.

3.6.3. Training

SOUL models are trained to maximise the likelihood. This optimization is carried out using a mini-batch version of Stochastic Back-propagation (see [24, 26] for more details). However, given the computational cost of each training batch, training corpora are usually resampled at each epoch: instead of performing several epochs over the whole training data, a different small random subset is used at each epoch.

To mitigate the impact of in-domain and out-of-domain corpora, the target language model was trained using for each epoch a set of n-grams of which 75% were sampled from TED data, and the remaining 25% from Gigaword.

SOUL translation models were trained on bilingual tuples constructed from the word alignments of training corpora's sentence pairs. The mixing of training corpora was more complicated as TED corpus contains both manual and automatic transcriptions. In an attempt to narrow the gap between ASR and MT as mentioned in Section 3.3, we used TED auto corpus along with TED manual to train our translation models. To separately evaluate the impact of each corpus, three configurations were tested. The first two consisted in training models on TED manual and TED auto separately. In the third configuration, a mix of TED data (manual and auto concatenated) and Gigaword was used, where 75% of n-grams

Systems	dev	test
Before SOUL	23.7	27.7
Adding all 4 SOUL TMs		
+ TMs TED manual	24.1	27.9
+ TMs TED auto	24.2	28.0
+ TMs mixing TED-GIGA	24.4	27.9
Adding all 4 SOUL TMs and SOUL target LM		
+ TMs TED manual + LM	24.3	27.9
+ TMs TED auto + LM	24.3	27.6
+ TMs mixing TED-GIGA + LM	**24.4**	**28.3**

Table 7: Results of the reranking process with various added feature functions. The first line indicates the result for the best MT system before SOUL. The upper and lower parts of the table show results of adding SOUL TMs and target LM into this system.

used at each epoch were sampled from the former, and 25% from the latter.

Table 7 presents results of adding SOUL features into the best MT system. The performance is evaluated in terms of BLEU scores on the automatically transcribed development and test sets. As shown in the upper part of the table, the models trained on TED auto yield slightly better results than those trained on TED manual. It might be due to the fact that hypotheses in the development and test sets were generated using source sentences automatically transcribed as described previously, and hence are closer to TED auto's bilingual tuples. However, the use of SOUL target language model gave gain only on the configuration trained on the mixed corpora of TED and Gigaword; the best result shown in the last line corresponds to the final system submitted for the evaluation as our primary system.

4. Conclusions

In this paper, we described our submissions for the IWSLT 2014 speech translation task. Our contribution is twofold: first, we investigated different approaches to adapt a standard speech recognition system to TED talks; then the different components of the MT system were improved for a better interaction with ASR output. The MT systems were trained using our in-house translation system (NCODE). We experimented with various techniques for bringing the ASR output data and the expected MT input data format as close as possible. In particular, number normalization and punctuation insertion both allowed to improve the translation quality over the baseline system on ASR data. We also exprimented with various configurations for including the ASR data as part of the MT system so as to adapt this system to the errors and other specific features of the speech recognition output.

Our best submission used both manual and ASR data pooled together for building one translation table. This system was augmented with the integration of continuous space models in a n-best rescoring step. Surprisingly, the gains on the ASR output test data were rather small as compared to the improvement obtained on very similar task for text translation (see [29, 25]). Further analyses are required to better explain these results.

5. Acknowledgements

The authors would like to expresses their gratitude to Jan Niehues for his help and advice in the preparation of this submission.

6. References

[1] J.-L. Gauvain, L. Lamel, and G. Adda, "The LIMSI Broadcast News Transcription System," *SPCOM*, vol. 37, no. 1-2, pp. 89–108, 2002.

[2] ——, "Partitioning and transcription of broadcast news data," *ICSLP*, vol. 98, no. 5, pp. 1335–1338, 1998.

[3] H. Hermansky, "Perceptual linear prediction (PLP) analysis for speech," *Journal of the Acoustical Society of America*, vol. 87, pp. 1738–1752, April 1990.

[4] P. Schwarz, P. Matějka, and J. Černocký, "Towards lower error rates in phoneme recognition," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science, P. Sojka, I. Kopeček, and K. Pala, Eds. Springer Berlin Heidelberg, 2004, vol. 3206, pp. 465–472.

[5] P. Fousek, L. Lamel, and J.-L. Gauvain, "On the use of mlp features for broadcast news transcription," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science, no. 5246/2008. Springer Verlag, Berlin/Heidelberg, 2008, pp. 303–310.

[6] Q. Zhu, A. Stolcke, B. Y. Chen, and N. Morgan, "Using mlp features in SRI's conversational speech recognition system." in *Interspeech*, 2005, pp. 2141–2144.

[7] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[8] L. Lamel, S. Courcinous, J. Despres, J.-L. Gauvain, Y. Josse, K. Kilgour, F. Kraft, V. B. Le, H. Ney, M. Nußbaum-Thom, I. Oparin, T. Schlippe, R. Schlüter, T. Schultz, T. F. da Silva, S. Stüker, M. Sundermeyer, B. Vieru, N. T. Vu, A. Waibel, and C. Woehrling, "Speech Recognition for Machine Translation in Quaero," in *IWSLT*, San Francisco, CA, USA, 2011.

[9] L. Lamel, " Multilingual Speech Processing Activities in Quaero: Application to Multimedia Search in Unstructured Data," in *The Fifth International Conference: Human Language Technologies - The Baltic Perspective*, Tartu, Estonia, October 4-5 2012, pp. 1–8.

[10] F. Casacuberta and E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30, no. 3, pp. 205–225, 2004.

[11] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. Fonollosa, and M. R. Costa-Jussà, "N-gram-based Machine Translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[12] J. M. Crego and J. B. Mariño, "Improving statistical MT by coupling reordering and decoding," *Machine Translation*, vol. 20, no. 3, pp. 199–215, 2006.

[13] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *KI-2002: Advances in artificial intelligence*, ser. LNAI, M. Jarke, J. Koehler, and G. Lakemeyer, Eds., vol. 2479. Springer Verlag, 2002, pp. 18–32.

[14] C. Tillmann, "A unigram orientation model for statistical machine translation," in *Proceedings of HLT-NAACL*, 2004, pp. 101–104.

[15] J. M. Crego, F. Yvon, and J. B. Mariño, "N-code: an open-source bilingual N-gram SMT toolkit," *Prague Bulletin of Mathematical Linguistics*, vol. 96, pp. 49–58, 2011.

[16] D. Déchelotte, G. Adda, A. Allauzen, H. Bonneau-Maynard, O. Galibert, J.-L. Gauvain, P. Langlais, and F. Yvon, "LIMSI's statistical translation systems for WMT'08," in *Proceedings of the Third Workshop on Statistical Machine Translation*, Columbus, Ohio, 2008, pp. 107–110.

[17] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, ser. SETQA-NLP '08, 2008, pp. 49–57.

[18] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *International Conference on New Methods in Language Processing*, Manchester, UK, 1994, pp. 44–49.

[19] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proceedings of ICSLP*, vol. 2, Denver, USA, 2002, pp. 901–904.

[20] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna, "Findings of the 2014 workshop on statistical machine translation," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, MA, June 2014, pp. 12–58.

[21] A. Allauzen, G. Adda, H. Bonneau-Maynard, J. M. Crego, H.-S. Le, A. Max, A. Lardilleux, T. Lavergne, A. Sokolov, G. Wisniewski, and F. Yvon, "LIMSI @ WMT11," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, 2011, pp. 309–315.

[22] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling punctuation prediction as machine translation," in *International Workshop on Spoken Language Translation (IWSLT 2011)*, 2011, pp. 238–245.

[23] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 1137–1155, 2003.

[24] H. Schwenk, D. Déchelotte, and J.-L. Gauvain, "Continuous space language models for statistical machine translation," in *Proceedings of the COLING/ACL on Main conference poster sessions*, Morristown, NJ, USA, 2006, pp. 723–730.

[25] A. Allauzen, N. Pécheux, Q. K. Do, M. Dinarelli, T. Lavergne, A. Max, H.-s. Le, and F. Yvon, "LIMSI @ WMT13," in *Proceedings of the Workshkop on Statistical Machine Translation*, Sofia, Bulgaria, 2013, pp. 62–69.

[26] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language model," in *Proceedings of ICASSP*, 2011, pp. 5524–5527.

[27] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous space translation models with neural networks," in *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, Montréal, Canada, June 2012, pp. 39–48.

[28] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012, pp. 427–436.

[29] T. Lavergne, A. Allauzen, H. S. Le, and F. Yvon, "LIMSI's experiments in domain adaptation for IWSLT 11," in *International Workshop on Spoken Language Translation, IWSLT*, 2011, pp. 62–67.

The NICT ASR System for IWSLT 2014

Peng Shen, Xugang Lu, Xinhui Hu, Naoyuki Kanda, Masahiro Saiko, Chiori Hori

Spoken Language Communication Laboratory,
National Institute of Information and Communications Technology,
Kyoto, Japan

peng.shen@nict.go.jp

Abstract

This paper describes our automatic speech recognition system for IWSLT2014 evaluation campaign. The system is based on weighted finite-state transducers and a combination of multiple subsystems which consists of four types of acoustic feature sets, four types of acoustic models, and N-gram and recurrent neural network language models. Compared with our system used in last year, we added additional subsystems based on deep neural network modeling on filter bank feature and convolutional deep neural network modeling on filter bank feature with tonal features. In addition, modifications and improvements on automatic acoustic segmentation and deep neural network speaker adaptation were applied. Compared with our last year's system on speech recognition experiments, our new system achieved 21.5% relative improvement on word error rate on the 2013 English test data set.

1. Introduction

TED talks are presentations to audience with wide topics related to Technology, Entertainment and Design (TED) in spontaneous speaking style [1]. Automatically transcribing TED talks with automatic speech recognition (ASR) technique is still a challenging task. The difficulties are due to the large variations of TED speech caused by many factors, for example, variations caused by disfluency, emotion, noise distortions, as well as variations caused by accent and ages of speakers. In this paper, we describe our ASR system for the English TED ASR track of the 2014 IWSLT evaluation campaign.

The system is a further development of our 2012 and 2013 ASR systems which utilized lots of state of the art technologies [2, 3]. An overview of our ASR system is depicted in Figure 1. In this figure, there are several processing blocks. The test TED talks were provided without any acoustic segmentation information. For convenience of processing and decoding, an automatic acoustic segmentation was first applied. Based on the segmentation, acoustic features were extracted. Next, decoding was applied on four types of acoustic models to produce decoding lattices, and rescoring was used on the N-best lists generated by the lattices. Based on the N-best lists, a ROVER processing was used to get the first pass ROVER result. Based on the first pass ROVER result, the language model adaptation and acoustic model adaptation were done. Then decoding and rescoring were done again with the adapted LM and acoustic models. Furthermore, the second pass ROVER was conducted. The adaptation, decoding, rescoring, and ROVER were done for several rounds.

Compared with the system we used in last year, new contributions are (1) refined acoustic segmentation algorithm; (2) deep neural network (DNN) acoustic model trained based on new types of acoustic features; (3) convolutional DNN (CNN-DNN) acoustic model trained based on filter bank feature concatenating with pitch feature. Besides these main changes, several other modifications were also added which showed performance improvement.

The rest of this paper is organized as follows. Sections 2 and 3 introduce the acoustic modeling and the language modeling. Section 4 describes the automatic acoustic segmentation algorithm. Section 5 introduces the decoding processing which includes LM rescore and N-best ROVER procedures. Experimental results as well as discussion of the results are given in Section 6. Conclusion is given in Section 7 .

2. Acoustic Modeling

2.1. Training Corpus

Three types of data corpus were used in training the acoustic models (as shown in table 1). 81.1 hours of Wall Street Journal (WSJ), 62.9 hours of HUB4 English Broadcast news which obtained from the Linguistic Data Consortium, and 167.8 hours of processed 760 TED talks crawled from its online web site published before 2011 (with SailAlign software for extracting text-aligned acoustic segments). WSJ is read speech, HUB4 is spontaneous broadcast news speech and TED is lecture style speech.

2.2. Feature Extraction

Four types of acoustic feature sets were extracted to build acoustic models. The first type of feature set is Mel-frequency cepstral coefficient (MFCC), which was extracted with a 25 ms Hamming window that was shifted at 10 ms intervals. The MFCC feature consisted of 12 MFCCs, logarithmic power (log-power), and their first and second or-

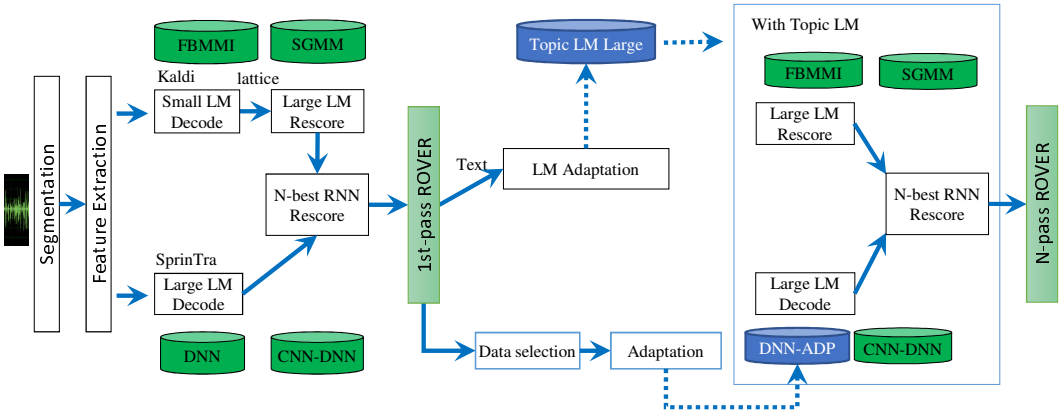


Figure 1: Overview of the NICT ASR system.

der derivatives. The dimensions of the acoustic feature vectors were 39. Then 7 adjacent frames were concatenated (3 on each side of the current frame) to make context dependent feature vectors. By applying a linear discriminate analysis (LDA), the concatenated feature vector was compressed to 40 dimensions. The 40-dimension vector was further decorrelated with a maximum likelihood linear transformation (MLLT). In addition, a feature space maximum likelihood linear regression (fMLLR) was also applied in speaker adaptive training (SAT) stage. The second type of acoustic feature set is a perceptual linear predictive cepstrum (PLP) feature, the same procedures as done on the MFCC feature were applied. The third type of feature set is log Mel filter bank feature or FBANK features. It has been shown to improve the performance of DNN based acoustic modeling than MFCC feature [4]. The fourth type of acoustic feature set is combination of FBANK feature with tonal feature. Although English is not a tonal language, it showed improvement in DNN modeling if tone feature is incorporated as an additional feature in acoustic modeling [5]. In both the third and fourth types of feature sets, the first and second time derivations of these features were utilized in acoustic modeling. In addition, because these two types of feature sets were used in different DNN architectures, their FBANK feature dimensions were also different. This will be explained in acoustic model training in next subsection.

2.3. Acoustic models and training

Four types of acoustic models were built in our system, they were FBMMI, SGMM, DNN and CNN-DNN acoustic models. To train these models, we first trained a basic context dependent triphone HMM model with GMM output probability. The final acoustic model has 7922 triphone tied states with 160180 Gaussian components. For improving the basic model, we further applied feature space maximum likelihood linear regression (fMLLR) for speaker adaptive training. This SAT-HMM/GMM model was used as a baseline for FBMMI, SGMM, DNN and CNN-DNN acoustic model training.

The FBMMI is a discriminative training with feature space boosted maximum mutual information (FBMMI) criterion [6]. The subspace GMM (SGMM) model was trained by clustering the Gaussians from the triphone HMM/GMM model. In addition, the FBMMI was also conducted on SGMM for discriminative training. The FBMMI and SGMM acoustic models were trained by two types of feature sets, MFCC and PLP. Therefore, four acoustic models were obtained.

Two types of DNN architectures were used for acoustic modeling. One is feedforward DNN (hereafter it is called as DNN). Another is convolutional DNN in which the input layer is with convolutional operator while other layers are feedforward DNN (hereafter it is called as CNN-DNN). In DNN training, a frame-based cross-entropy criterion was first applied in the first stage, then a sequential discriminative training based on a state level minimum Bayesian risk criterion (sMBR) was adopted for the second stage training [7]. In CNN-DNN training, only the frame-based cross-entropy criterion was used. For training the DNN and CNN-DNN, different types of feature sets were used. For MFCC and PLP feature sets, the DNN architecture was configured as: 300-2100*5-7922, i.e., input layer was with 300 neurons, 5 hidden layers with 2100 neurons for each, and 7922 neurons in the output layer. The input layer feature was transformed by LDA from 15 consecutive frames of either MFCC or PLP feature (from SAT-HMM/GMM model). For FBANK feature used in DNN, 24 Mel filter banks were used (hereafter as FBANK24 feature type). The DNN was configured as: 1080-2100*5-7922.

In CNN-DNN modeling, compared with DNNs, CNN restricts the network architecture with local connections and weight sharing so that it can explores local correlation in feature processing [8]. Our CNN-DNN has one convolutional layer with convolution and polling operations. The configuration of the convolutional layer as: 128 filters with filter

Table 1: Details of acoustic model training data

Corpus	Hours	Type	Data
WSJ	81.1	Read	LDC93S6B, LDC94S13B
HUB4	62.9	Broadcast	LDC97S44, LDC98S71
TED	167.8	Lecture	760 talks (Before 2011)

size and shift as 9 and 1 for each. In the pooling layer, local averaging and sub-sampling were performed to reduce the resolution of the feature map and the sensitivity of the output to input shifts and distortions. The pooling width and shift was set to 2 and 2, respectively. The output from the pooling layer was further processed with feedforward DNN with 4 hidden layers (2100 neurons in each layer), and one output layer (7922 neurons). In training the CNN-DNN, FBANK feature with tone feature set was used. 40 Mel filter banks and 3 dimensional tone features were used (hereafter as FBANK40+Pitch feature type).

2.4. Speaker Adaptation for DNN

In our system, speaker adaptation on DNN AMs were applied. The adaptation was operated on the third hidden layer of the DNNs based on our previous work [9]. The adaptation data was selected based on word confidence from decoding results (confidence threshold 0.7 was chosen in our study). Different from last year's adaptation processing, the adaptation data was selected based on the ROVER result. In order to overcome the overtraining problem in adaptation, a L2 regularization on the model parameters was utilized. 4 rounds adaptation were conducted on the DNN models. In each adaptation, the learning rate was set to 0.001, the number of training epoches were set to 20.

3. Language Modeling

3.1. Training data

Table 2 shows the data for training language models. It contains two categories of textual corpora that are allowed by the IWSLT evaluation campaign. One is in-domain corpus TED talk transcripts supplied by the IWSLT2014 committee, another are out-of-domain corpora. For the out-of-domain corpora, News Commentary V7 and Europarl V6 provided by the IWSLT2014 committee were used for LM training without selections, but English Gigawords and News Shuffle were further selected for the training. All of these data were normalized (or pre-processed) by using a non-standard-word expansion tools [10], so that all those non-standard words such as abbreviation, numbers etc, were converted to simple words. For examples, words "CO2" and "95%" were converted to "CO two" and "ninety five percent." Duplicated sentences were removed during this normalization process.

Table 2: Training data of language models.

Category	Corpus	Tokens
In-domain	TED Talks	3.2M
Out-of domain	NewsCommentary V7	4.6M
	Europarl V7	50.0M
	English Gigawords 5th ed.	2.7G
	News Shuffle	732.8M

3.2. Domain adapted n-gram LM

The first pass of speech decoding was performed using a domain adapted n-gram LM. The adapted LM was built by interpolating a in-domain n-gram and several adaptation n-grams. The in-domain n-gram was constructed by using the in-domain data, and the adaptation n-grams were constructed by using the selected sentences from the out-of-domain corpora. However, since there are many sentences in the out-of-domain that are highly mismatched to the TED domain, these sentences will be harmful to LM if they are added to training data. Therefore, we built adaptation LMs by selecting adequate training sentences from two of the out-of-domain corpora - English Gigawords and News Shuffle. Since the News Commentary data and the Europarl are relative small, no selection was conducted on them.

The sentence selection was based on a cross-entropy difference metric [11] which was biased towards sentences that were both similar to the in-domain data and unlike the average of the out-of-domain data. Here, the similarity and unlikeness were measured by the sentence entropy (or perplexity) with respect to in-domain LM and out-of-domain LM, respectively. Detailed description about this selection algorithm can be referred in [12]. Finally, about 30.0M sentences (560M tokens) from the English Gigaword data, and 7.6M sentences (143.8M tokens) were selected.

Using the SRILM toolkit [13], the modified Kneser-Ney smoothed n-grams (n=4) were constructed for in-domain LM using the TED corpus, and for adaptation LMs accordingly using the selected sentences, the News Commentary V7 data and the Europarl V7 data. The domain adapted LM was achieved by linearly interpolating these n-grams, with the development set defined in the IWSLT evaluation campaign for optimization. In all these training process, a vocabulary of 123K words from the CMU Pronunciation Dictionary [14] and the TED corpus was used.

3.3. Topic adapted n-gram LM

The second pass of speech decoding was conducted using a topic adapted LM constructed by the recognition results of the first decoding pass. The sentence selection for the topic adapted LM was conducted in the same way as for the domain adapted LM. The data sources for selection were still the English Gigawords and the News Shffled, however, the recognition results obtained from the first decoding pass were

used as the seed data for selection. The sentence cross entropy was measured between two n-gram LMs, one was built by using recognition results of all talks in the first decoding pass, another was built by using 2000 sentences randomly selected from the resource data. Finally, 61.7M sentences (246.7M tokens) were selected from the English Gigawords, and 3.8M sentences (65.7M tokens were selected from the News Shuffle. Two n-grams (n=4) were built by using these sentences individually. The topic adapted LM was then constructed by linearly interpolating these two LMs, other two LMs built respectively by the News Commentary and Europarl, (for these two corpora, no sentence selections are conducted with them), and the in-domain LM.

3.4. RNNLM

In this system, N-best list rescoring was adopted and performed using a recurrent neural network(RNN) LM [15]. In our RNN, the number of units in the hidden layer and classes in the output layer were 480 and 300, respectively. Back-propagation Through Time (BPTT) with truncated time order 5 was used in RNN training. The training data for the RNN was the same as that for the domain adapted n-gram LM described above. To decrease the training time, only one-tenth of the selected out-of-domain sentences were used for the training.

4. Automatic Segmentation

In this year's evaluation, the whole TED talks in the test data set were provided without any acoustic segmentation information. For convenience of decoding and rescoring, acoustic segmentation was first done. A combination method of a voice activity detection (VAD) algorithm and acoustic event detection (AED) algorithm was utilized for this purpose. In designing the VAD algorithm, signal power energy and spectral centroid features were used. In AED, five GMMs corresponding to five acoustic events (speech, music, applause, laugh and background noise) mostly appeared in lecture speech were trained in this study. MFCC feature was used in GMM training, and the diagonal GMM consists 16 mixtures was used in AED. The acoustic segmentation was done based on merging the detection results of VAD and GMM. In merging, a hang-over scheme with minimum durations of non-speech event as 800ms, and minimum duration of speech event as 160ms was applied. Based on the segmented utterances, the feature extraction, decoding and ROVER were carried out in recognition experiments.

5. Decoding and ROVER

5.1. Decoding System

Two types of WFST-based decoders were used. One is Kaldi decoder, the other is NICT SprinTra decoder. The Kaldi decoder was used for FBMMI and SGMM acoustic model based decoding, and SprinTra decoder was utilized for DNN and CNN-DNN acoustic model based decoding.

In decoding with Kaldi decoder, a small 4-gram LM was first used to produce word lattice. Then a large 4-gram LM was applied for rescoring on the word lattice. For improving the performance, the RNN LM was further applied on the N-best list for rescoring. In decoding with NICT SprinTra decoder, the large 4-gram LM was directly used. Based on the decoding word lattice, RNN LM was also used on the N-best list for rescoring.

5.2. N-best ROVER

A N-best recognizer output voting error reduction (ROVER) algorithm was applied to combine all the subsystems for further improving the performance. This year, subsystems with four types of acoustic models (FBMMI, SGMM, DNN and CNN-DNN) and four types of feature sets (MFCC, PLP, FBANK24, FBANK40+Pitch) were combined in ROVER processing. For each subsystem, 50-best lists from 4-gram LM and RNN LM rescoring processing were used. In ROVER, the combination weights were selected based on our experimental results on the development data set.

6. Experimental Results

6.1. DNN Speaker Adaptation

The algorithm of speaker adaptation used in this year is similar to last year's system. But the adaptation data selection is different from last year's system. In last year's system, the adaptation data set was picked up based on the DNN decoding result. Considering that ROVER result is always better than one of the DNN decoding result, the adaptation data was selected based on the ROVER result in this year. For comparison of the two adaptation data selection methods, we showed the results in Figure 2. The decoding/rescoring results are also included for comparison. In our 2013 system, after the first pass ROVER, topic adaptation on LM was conducted. With the adapted LM, we could obtain 0.4% improvement for both 2011 and 2012 test data sets on DNN based decoding. Then the adaptation data was selected based on this DNN decoding result. In this year, we simply changed the adaptation data selection method based on word confidence calculated in the ROVER step. From the decoding results, 0.7% and 0.9% improvements were obtained for 2011 and 2012 test data sets, respectively. With this new process, our speaker adaptation on DNN can be done for multiple rounds for obtaining better performance. Table 3 shows the results of N-rounds DNN speaker adaptation process. The results showed that consistent improvements were obtained with 4-rounds DNN adaptation for each feature set separately. However, no further improvement was obtained for ROVER result with fifth round adaptation.

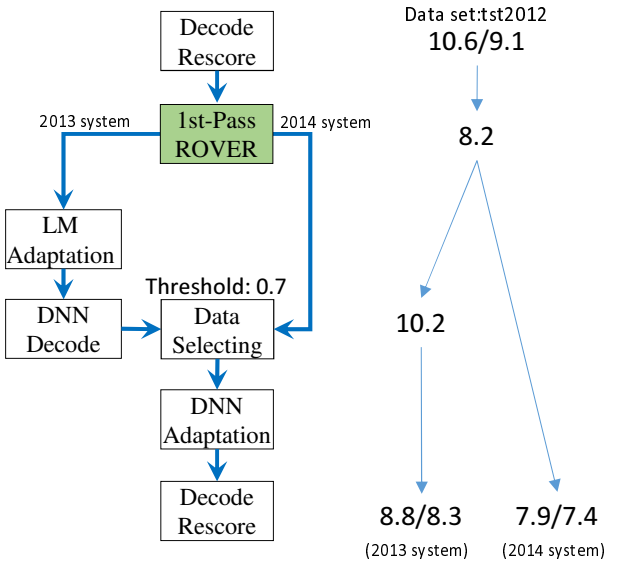


Figure 2: The adaptation process and evaluation results (WER %) of 2013, 2014 system. The feature of decoding/rescoring results are MFCC, The ROVER consists of FBMMI, SGMM, DNN acoustic models with feature MFCC and PLP.

Table 3: Contribution of N-rounds adaptation; Decoding/rescoring results are listed for DNN with feature MFCC, PLP and FBANK; The ROVER consits of all the subsystems.

subsystem	MFCC	PLP	FBANK	ROVER
DNN-baseline	16.0/14.8	16.3/15.3	15.2/14.6	12.7
1st round	12.3/11.8	12.3/11.9	12.7/12.2	11.6
2nd round	11.7/11.4	11.7/11.4	11.9/11.6	11.2
3rd round	11.5/11.3	11.4/11.2	11.6/11.4	11.1
4th round	11.3/11.2	11.3/11.1	11.4/11.3	11.1

6.2. Searching Beam and ROVER Weights

Increasing the searching beam width in decoding always helps to improve the performance but at the cost of increasing searching time. In our experiments, we set beam width to 13 (the same as in last year) for both Kaldi and SprinTra decoders in the first few steps of decoding. In the last DNN adaptation step, the beam width was set to 17 which resulted in 0.1% improvement in the WER.

In ROVER processing, the combination weights were set as 1:1:2 for FBMMI, SGMM and DNN in last year. After adding the DNN-FBANK24 and CNN-DNN acoustic model based subsystems, the combination weights were re-investigated based on the development data set. Different combination weight sets were set for ROVER: 1:1:3:3 for FBMMI, SGMM, DNN, CNN-DNN for the first pass ROVER and 1:1:7:1 for N-round pass ROVER (N=2,3,4,5).

6.3. Contributions of Subsystems

Table 4 shows the results on 2013 test data set with different combinations of subsystems in the first pass ROVER. With

Table 4: Contribution of each subsystems(first pass ROVER); data set: 2013 test data set

subsystem	sys1	sys2	sys3	sys4	sys5	sys6	sys7	sys8
FBMMI	○		○		○	○		○
SGMM	○		○		○	○		○
DNN-mfcc		○	○	○	○	○	○	○
DNN-plp		○	○	○	○	○	○	○
DNN-fbank				○		○	○	○
CNN-DNN					○		○	○
WER(%)	18.1	14.5	13.8	13.4	13.1	13.1	12.9	12.7

Table 5: Contribution of each subsystems(the fifth pass ROVER), with topic adapted LM and speaker adaptation for DNN models; data set: 2013 test data set

subsystem	sys1	sys2	sys3	sys4	sys5	sys6	sys7	sys8
FBMMI	○		○		○	○		○
SGMM	○		○		○	○		○
DNN-mfcc		○	○	○	○	○	○	○
DNN-plp		○	○	○	○	○	○	○
DNN-fbank				○		○	○	○
CNN-DNN					○		○	○
WER(%)	17.8	11.1	11.1	11.0	11.1	11.1	11.1	11.0

the subsystems used in last year (sys3), we obtained 13.8% WER. 1.1% absolute improvement was obtained after adding the DNN-FBANK24 and CNN-DNN based subsystems in ROVER. Also from this table, we can see that although DNN and CNN-DNN subsystems obtained quite low WER, taking the FBMMI and SGMM based subsystems in ROVER processing still helped to improve the performance (about 0.2% improvement).

Table 5 shows the results of the fifth pass ROVER step. In this step, the LM and DNN acoustic model were adapted with the methods described in the previous section. Different to the first pass ROVER result, we obtained almost the same result by only combing DNN acoustic model based subsystems with or without the FBMMI, SGMM and CNN-DNN based subsystems.

6.4. Summary of Results

Table 6 shows the summary of our ASR system comparing with last year's official best rest for 2011, 2012, and 2013 test data sets. Compared to last year's official result, this year ASR approach achieved a better performance. The automatic segmentation, combination of new subsystems in ROVER, and multi-rounds speaker adaptation contributed the most of the improvements. After 4-rounds speaker adaptation on DNN acoustic models, there was no further improvement in final ROVER processing. For this year's test set, we obtained 8.4% WER.

Table 6: The final results (WER %) of the test sets: 2011, 2012, 2013 and 2014.(* means using NICT's references)

	tst2011	tst2012	tst2013	tst2014
Official best(2013)	7.9	8.6	13.5	-
NICT 2014	6.5*	7.0*	10.6	8.4

7. Conclusions

In this study, we describe our ASR system for the IWSLT 2014 evaluation campaign. Our ASR system consists of four types of acoustic models (FBMMI, SGMM, DNN and CNN-DNN), four types of acoustic features (MFCC, PLP, FBANK24 and FBANK40+Pitch), and two types of LMs (N-gram and RNN). Several improvements were conducted, such as new acoustic models, automatic segmentation, and DNN speaker adaptation. The results of our proposed approaches demonstrate a better performance than that of last year.

8. References

[1] TED, http://www.ted.com/

[2] H. Yamamoto, Y. Wu, C. Huang, X. Lu, P. R. Dixon, S. Matsuda, C. Hori, and H. Kashioka, "The NICT ASR System for IWSLT2012," In *Proc. of IWSLT*, 2012.

[3] C. Huang, P. R. Dixon, S. Matsuda, Y. Wu, X. Lu, M. Saiko, and C. Hori, "The NICT ASR system for IWSLT 2013," In *Proc. of IWSLT*, 2013.

[4] L. Deng and J. Li and J-T. Huang, K. Yao, D. Yu, F. Seide,M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero, "Recent advances in deep learning for speech research at Microsoft," In *Proc. of ICASSP*, 2013.

[5] X. Lei, M-Y Hwang, and M. Ostendorf, "Incorporating tone-related MLP posteriors in the feature representation for Mandarin ASR," In *Proc. Eur. Conf. Speech Communication Technology*, 2005.

[6] D. Povey, S. M. Chu, J. Pelecanos, and H. Soltau, "Approaches to Speech Recognition based on Speaker Recognition Techniques," Chapter in forthcoming GALE book.

[7] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. of Interspeech*, 2013.

[8] X. Hu, X. Lu and, and C. Hori, "Mandarin Speech Recognition Using Convolution Neural Network With Augmented Tone Features," In *Proc. of ISCSLP*, 2014.

[9] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training using deep neural networks," in *Proc of ICASSP*, Italy, 2014.

[10] R. Sproat, A. W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards, "Normalization of Non-Standard Words," in Computer Speech and Language , pp.287-333, 2001.

[11] R. C. Moore, and W. Lewis, "Intelligent Selection of language Model Training Data," in *Proc. of ALC*, 2010.

[12] P. Bell, H. Yamamoto, P. Swietojanski, Y. Z. Wu, F. McInnes, C. Hori, and S. Renals, "A Lecture Transcription System Combining Neural Network Acoustic and Language Model," in *Proc. of Interspeech*, 2013.

[13] A. Stolcke, "SRILM - An extensible Language Modeling Toolkit," in *Proc. of ICSLP*, 2002.

[14] http://www.speech.cs.cmu.edu/cgi-bin/cmudic

[15] T. Mikolov, M. Cettolo, L. Burget, J. Cernnokcy, and S. Khudanpur, "Recurrent Neural Network Based Language Model," in *Proc. of Interspeech*, 2010.

The KIT Translation Systems for IWSLT 2014

Isabel Slawik, Mohammed Mediani, Jan Niehues, Yuqi Zhang,
Eunah Cho, Teresa Herrmann, Thanh-Le Ha and Alex Waibel

Institute for Anthropomatics and Robotics
KIT - Karlsruhe Institute of Technology, Germany
firstname.lastname@kit.edu

Abstract

In this paper, we present the KIT systems participating in the TED translation tasks of the IWSLT 2014 machine translation evaluation. We submitted phrase-based translation systems for all three official directions, namely English→German, German→English, and English→French, as well as for the optional directions English→Chinese and English→Arabic. For the official directions we built systems both for the machine translation as well as the spoken language translation track.

This year we improved our systems' performance over last year through n-best list rescoring using neural network-based translation and language models and novel preordering rules based on tree information of multiple syntactic levels. Furthermore, we could successfully apply a novel phrase extraction algorithm and transliteration of unknown words for Arabic. We also submitted a contrastive system for German→English built with stemmed German adjectives.

For the SLT tracks, we used a monolingual translation system to translate the lowercased ASR hypotheses with all punctuation stripped to truecased, punctuated output as a preprocessing step to our usual translation system.

1. Introduction

The Karlsruhe Institute of Technology participated in the IWSLT 2014 Evaluation Campaign with systems for English→German, German→English and English→French, covering all official directions, as well as English→Chinese and English→Arabic. All systems were submitted for the machine translation (MT) track, with additional systems for the spoken language translation (SLT) track in the official directions. This year we also submitted three contrastive systems in order to directly compare the impact of some of our new models. For English→German we focused on the impact of rescoring on our system, for German→English we submitted a contrastive system that was built with stemmed adjectives on the German source side, and for English→Arabic we compared our alternative phrase table pruning method to the standard approach.

We focused our efforts on five components this year. The handling of ASR input was further refined (Section 3), and we newly implemented Restricted Boltzmann Machine (RBM)-based translation and language models for rescoring (Section 4), an alternative method to prune the phrase table (Section 5), a method to transliterate unknown words into Arabic (Section 6) and multiple level tree-based (MLT) reordering rules (Section 7).

The following section briefly describes our baseline system, while Sections 3 through 7 present the different components and extensions used by our phrase-based translation systems. After that, the results of the different experiments for the five language pairs we participated in are presented in Section 8 before we summarize our findings in Section 9.

2. Baseline system

All our systems are phrase-based systems. With the exception of the English→Chinese system, they are trained on the provided EPPS, NC and TED corpora. We also used the provided Common Crawl corpus for English↔German and Giga for English→French. For the monolingual training data we used the target side of all bilingual corpora as well as the News Shuffle corpus. Additionally, we included the Gigaword corpus for English→French and German→English. The English→Chinese system setup is described in Section 8.5.

Before training and translation, the data is preprocessed. During this phase, exceedingly long sentences and sentence pairs with a large length difference are discarded from the training data. We normalize special dates, numbers and symbols and smart-case the first letter of every sentence. For German→English, we split up compounds [1] on the source side of the corpus. Since the Common Crawl and Giga English→French corpus are very noisy, we trained an SVM classifier to filter them as described in [2].

After preprocessing, the parallel corpora are word-aligned using the GIZA++ Toolkit [3] in both directions. The resulting alignments are then combined using the grow-diag-final-and heuristic. The phrases are extracted using the Moses toolkit [4] and then scored by our in-house parallel phrase scorer [5]. Phrase table adaptation combining an in-domain and out-of-domain phrase table is performed as described in [6]. All translations are generated using our in-house phrase-based decoder [7].

Unless stated otherwise, we used 4-gram language mod-

els with modified Kneser-Ney smoothing, trained with the SRILM toolkit [8] and scored in the decoding process with KenLM [9]. In addition to common word-based language models, we used two token-based language models. The bilingual language model is used to increase the bilingual context during translation beyond phrase boundaries as described in [10]. A token consists of a target word and all its aligned source words. As a second token language model, we use a cluster language model based on word classes. This helps alleviate the sparsity problem for surface words by replacing every word in the training corpus with its cluster ID calculated by the MKCLS algorithm [11].

We use two main reordering models in our systems. The first consists of automatically learned reordering rules based on POS sequences [12] and syntactic parse tree constituents [13, 14] and performs source sentence reordering according to target language word order [15, 16, 17]. The resulting reordering possibilities for each source sentence are then encoded in a lattice. The second model is a lexicalized reordering model [18] which stores reordering probabilities for each phrase pair.

As an additional model, we use a Discriminative Word Lexicon (DWL) using source context features as described in [19].

We tune our systems using Minimum Error Rate Training (MERT) against the BLEU score as described in [20].

3. Preprocessing for speech translation

A conventional automatic speech recognition (ASR) system generates a stream of recognized words without punctuation marks or reliable case information. Therefore, when we use the ASR output as input for our MT system, it does not fit the style and format of the training data. In order to perform special preprocessing on the SLT test data, we use a monolingual translation system as presented in [21]. The system inserts punctuation marks and corrects case information, so that there is less divergence between the MT training data and the SLT input data. As sentence boundaries are already given in the test sets, we leave them as they are but predict other punctuation marks within the segment. This preprocessing will be denoted as Monolingual Comma and Case Insertion (MCCI).

For building the systems, we took the preprocessed source side of the parallel training data. We remove all punctuation marks from the data and insert a final period at the end of each line. In addition to this, all words are lowercased. This data is used as the source side of our monolingual translation systems. For the target side of the monolingual translation system, we keep the punctuation marks as well as case information, so that the "translation" of our MCCI system consists of inserting punctuation marks and correcting case information.

We built an MCCI system for English and German and applied it to all three official SLT track directions English→German, German→English and English→French.

4. n-best list rescoring

We perform additional experiments to use a neural network language and translation model in n-best list rescoring.

We train an 8-gram Restricted Boltzmann Machine (RBM)-based language model [22] on the in-domain TED corpus. The language model uses 32 hidden units and a shared word representation with 512 dimensionsUnigram sampling is applied as described in [23].

In addition, we use an RBM-based translation model inspired by the work of Devlin et al. [24]. The RBM models the joined probability of 8 target words and a set of attached source words. The set of attached source words is calculated as follows: We first use the the source word aligned to the last target word in the 8-gram. If this does not exist, we take the source word aligned to the nearest target word. The set of source words consists then of this source word, its previous 5 source words and its following 5 source words.

We create this set of 8 target and 11 source words for every target 8-gram in the parallel in-domain TED corpus. In rescoring, we then calculate the free energy of the RBM given the 8-gram and its source set as input. The sum of all free energies in the sentence is used as an additional feature for rescoring.

The 300-best list of the test set is then rescored using the additional features. In order to train the weights for the original features as well as the RBM-based models, we use the ListNet algorithm [25]. We use stochastic gradient descent to find the best weights and use batched updates with a batch size of 10.

5. Alternative phrase table pruning

For efficiency reasons, we always perform a phrase table pruning before decoding. Basically, we use a log-linear model with some a-priori fixed weights in order to rank the different phrase table entries associated with a given source n-gram. The n-best entries are then selected (n being a fixed integer). In the Arabic system, we experimented with a slightly different model in order to rank the entries. The first difference to our standard is that the different features are pre-normalized. Based on other experiments (not reported in this paper), the ℓ^3-normalization is the best suited for this task. That is, each feature value is divided by the cubic root of the sum of all the values raised to the power of 3.

Another difference resides in the fact that the ranking is based on the distance between the phrase table entries and a reference entry. The latter is obtained by combining the maximum scores of the different features in one entry. Based on the same aforementioned experiments, we selected the Jensen-Shannon distance measure for this task [26].

6. Arabic transliteration

In most cases, untranslated words break the harmony of the translation into a language which uses a different scripting (e.g. English into Arabic.) Therefore, it is more conve-

س → s
ن → n
ب → b

Figure 1: Examples of trivial correspondences

nient to transliterate those untranslated words, as they are unlikely to hurt the system performance further. Our transliteration is mostly similar to the character-based translation in its transliteration part [27]. It is consequently a statistical phrase-based translation based on unigram characters.

The corresponding training data of this system is mainly a subset of the word pairs obtained from the aligned corpora (TED and UN). First, the Arabic word of each aligned pair is roughly transliterated into English, using only trivial correspondences (see Fig. 1 for an example). The Levenshtein distance ratio is then computed between the resulting rough transliteration and the English word. Finally, we retain only pairs with ratios higher than a certain threshold (our threshold was empirically set to 0.5).

7. Multi-level tree reordering rules

For our English-Chinese translation we applied a novel rule-based preordering approach [28], which uses the tree information of multiple syntactic levels. This approach extends the tree-based reordering [17] from one level into multiple levels, which has the capability to process complex reordering cases.

Reordering patterns are based on multiple levels of the syntax tree. Figure 2 illustrates how the reordering patterns are detected. The detection starts from the root node of the syntax tree, goes downwards multiple levels and uses the nodes in these levels to detect the reordering pattern. In this example, the nodes that are used for detecting the reordering pattern are colored gray and have an italic font. The leaf nodes in the syntax tree are the words in the sentence. According to the alignment information, the node labeled with *NP* should be moved to the first place in the translation and the node labeled with *IN of* needs to be moved to the second place in the translated sentence. So from the root node with a search depth of 3, the following reordering pattern can be found:

```
NP ( CD0 NP ( NP ( JJ1 NNS2 ) PP ( IN3
NP4 ) ) ) -> NP IN CD JJ NNS

-> 4 3 0 1 2 (alternative with index)
```

The algorithm for rule extraction detects the reordering patterns from all nodes in the syntax tree and it goes downwards for any number of levels, until it reaches the lowest level in the subtrees. The probability of the reordering patterns are calculated based on the frequency of their occurrences in the training corpus. In addition, reordering patterns that appear less often than a threshold are ignored in order to prevent too concrete rules lacking generalization capability and overfitting.

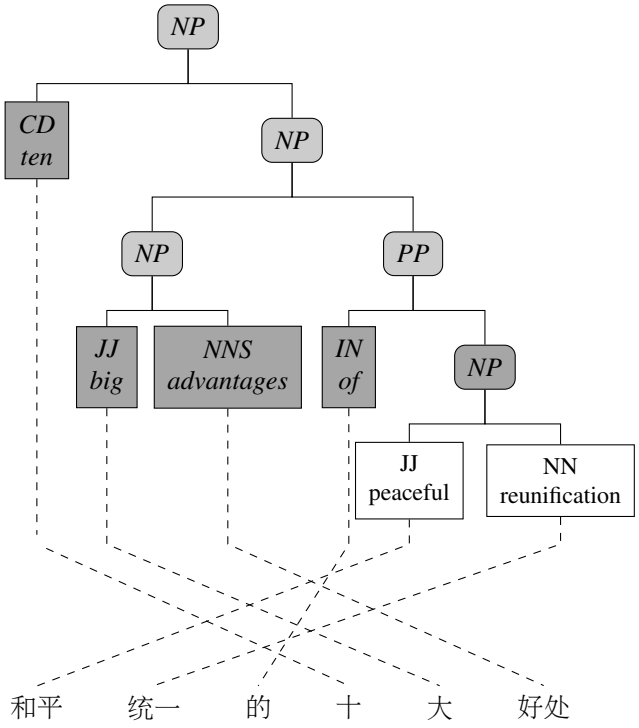


Figure 2: Detection of reordering pattern from multiple syntactic levels

When applying the rules prior to translation, the syntax tree is traversed by depth first search from the root of each subtree to its leaves. If a rule can be applied for a subtree at a given level, a new path for this reordering will be added to the word lattice for decoding. As long as rules can be applied on a subtree for a certain depth, the rules are applied and the search for rule application on this subtree stops.The search continues on the next subtree.

This multiple level tree-based (MLT) reordering rules can be combined with other types of reordering rules. This is done by combining the generated paths from different rules into one word lattice.

8. Results

In this section we present a summary of our experiments for both the MT and SLT tracks in the IWSLT 2014 evaluation. All the reported results are case-sensitive BLEU scores calculated on the provided development and test sets.

8.1. English→German

Table 1 shows the development stages of the English→German system. The baseline translation system uses two reordering models. First, in preprocessing, different possible source reorderings are encoded in a lattice. We used short-range and long-range POS-based reordering rules as well as tree-based rules. Secondly, a lexicalized reordering

model on the phrase level is used. The phrase table is adapted by combining two phrase tables, one trained on all training data and one trained only on the TED in-domain corpus. Furthermore, the translation process is modeled using a bilingual language model trained on all parallel data and a discriminative word lexicon trained on the TED corpus. The DWL uses source context features. Finally, five language models are used. Three are word-based models, the first of which is trained on all available German data. The second one is trained only on the TED corpus. Finally, we use a word-based model trained on 5M sentences chosen through data selection [29]. In addition, a 9-gram POS-based language model and a 9-gram cluster language model using 1000 MKCLS classes are used. Afterwards, we rescored the system using the weights trained using the ListNet algorithm described in Section 4. The rescoring was trained on the test2010 and test2011 data and dev2010 was used as a cross-validation set. This results in an improvement of 0.3 BLEU points. Then we added an RBM-based language model and an RBM-based translation model. We could improve by using the RBM-based translation model by 0.4 BLEU points, reaching the best BLEU score on test2012 with 24.31 BLEU points. This system was submitted as our primary system for English→German. The baseline system without rescoring was submitted as a contrastive system.

System	Dev	Test
Baseline	27.3	23.67
Rescoring	-	23.97
RBMLM	-	23.94
RBMTM	-	**24.31**

Table 1: Experiments for English→German (MT)

8.1.1. SLT track

Table 2 shows the translation quality of the individual system components. First we used the MT system and tested it on the SLT test set dev2010. After adding inter-sentence punctuation marks to the ASR hypothesis using the MCCI approach, we could improve by 1.3 BLEU points. Afterwards, we also used the ListNet-based rescoring for this task. This time we used only test2010 as a training set and test2011 as our cross-validation set. This improved the translation quality by 0.1 BLEU points. Finally, we added the RBM-based language model and translation model. This gave additional improvements of 0.1 BLEU points. We submitted the MCCI system as a contrastive system and the system using RBMLM and RBMTM in rescoring as our primary one.

8.2. German→English

Table 3 presents the results of our experiments for German→English. Our baseline system already incorporates a number of advanced models. Reordering is done using both

System	Dev	Test
Baseline	27.3	17.57
MCCI	-	18.83
Rescoring	-	18.91
RBMLM	-	19.02
RBMTM	-	18.96
RBMLM+TM	-	**19.01**

Table 2: Experiments for English→German (SLT)

POS-based preordering rules as well as a lexicalized reordering model. We adapted the in-domain and background phrase tables using the union candidate selection method. The system also includes a DWL trained on the in-domain data and five language models. In addition to the large background language model trained on all available English data, our baseline uses an in-domain language model, a background and in-domain bilingual language model, as well as a 9-gram in-domain cluster language model trained with 100 word classes. If we extend the preordering rules to include rules derived from parse trees, we can achieve a slight gain in BLEU. While the development score stays almost the same, we accomplish an improvement of nearly 0.3 BLEU points on the test data by extending the DWL to include source context. Training the DWL on n-best list data results in a similar gain in BLEU points yet again. We can further improve the score by applying the preordering rules learned from parse trees recursively. As our final model, we included a language model trained on on data automatically selected using cross-entropy differences [29]. We selected the top 10M sentences to train the language model. This leads to our final score of 31.98 BLEU points, almost 1 BLEU point over our baseline.

System	Dev	Test
Baseline	38.57	31.01
+ Tree Rules	38.79	31.04
+ DWL Source Context	38.78	31.32
+ DWL n-best List	38.86	31.63
+ Recursive Rules	38.92	31.71
+ Data Selection	39.03	**31.98**

Table 3: Experiments for German→English (MT)

8.2.1. Adjective stemming

Based on the system performing best in the previous experiment, we also submitted a contrastive system for German→English that employs stemming of adjectives.

Since German is a morphologically rich language, we are dealing with many surface forms. This creates data sparsity problems, as every surface form is treated as a distinct word in German. When translating into English, some of

System	Dev	Test
Primary	39.03	**31.98**
Stemmed	39.22	31.68

Table 4: Contrastive system for German→English (MT)

the information encoded in inflections such as gender or case may be discarded. However, stemming the whole German corpus hurts translation since too much information is lost. We therefore experimented with only stemming adjectives, which in German can have five different suffixes depending on the gender and case. The stemming was performed on the preprocessed files before compound splitting. The files were tagged with the TreeTagger [12] and the RFTagger [30]. We based our decision when and how to stem on the fine-grained tags output by the RFTagger. We only stemmed words tagged as an attributive adjective, since they are inflected in German. If the word as tagged as a comparative or superlative, we manually removed the inflected suffix in order to maintain the comparative nature of the adjective. For all other adjectives, we used the stem output by the TreeTagger. After stemming, compound splitting was applied as described in Section 2.

We then trained a new alignment and phrasetable on the stemmed corpora. Previous experiments had shown that using the stemmed phrasetable in conjunction with the unstemmed one gave better results than forcing the system to use the stemmed variant alone. However, our best system includes a DWL, biLM and cluster LM, which cannot be applied to the stemmed phrases in a straightforward manner. We therefore decided to unstem our phrasetable given the stems seen in the dev and test data. We looked at all the stem mappings from the development and test data and compiled a stem lexicon, mapping the surface forms observed in the Dev/Test data to their corresponding stems. We then applied this lexicon in reverse on our phrase table, in effect duplicating every entry containing a stemmed adjective with the inflected form replacing the stem. For translation we concatenated the default phrase table and the stemmed phrase table and combined the features log-linearly. This way our system was able to learn a weighing of the phrase scores during MERT. The resulting scores are reported in Table 4. While the stemmed system performs worse on the test data according to BLEU score, it does outperform our primary system on the development data. Using the stemmed system, we are able to translate seven adjectives we were not able to translate with our primary system. We therefore decided to submit our stemmed system as a contrastive system to fully evaluate our system's performance.

8.2.2. SLT track

Table 5 gives an overview of our systems for German→English SLT. As a baseline for the spoken language translation task, we used our best-performing system from the MT task. Applied to the ASR transcripts with only standard preprocessing, this gives us a baseline of 16.86 BLEU points. We can increase this score by nearly two BLEU points simply by adding a final period to every ASR segment. This shows that punctuation greatly influences the performance of our system. When we apply the more sophisticated MCCI system for punctuation and true casing of the test data, we achieve a similar improvement over the previous system. The last 0.2 BLEU points are gained by re-optimizing the system on development data that has been run through the MCCI system, resulting in our final system.

ASR Adaptation	Dev	Test
Baseline	39.03	16.86
+ period	-	18.79
MCCI	-	20.59
+ dev MCCI	35.79	**20.79**

Table 5: Experiments for German→English (SLT)

8.3. English→French

Table 6 summarizes the experiments performed for this direction.

The translation model of the baseline was built from TED, EPPS, NC, and Common-crawl corpora. It uses short-range POS-based reordering rules trained on TED, EPPS, and NC. It is also adapted to an in-domain translation model, exclusively trained on the TED corpus, using the union candidate selection method. In addition, 5 language models are used, 3 of which are conventional word-based LMs. One of the remaining LMs is a bilingual LM and the other is a cluster-based LM. The word-based LMs are trained on the French part of the parallel data, the monolingual data, and the union of all the French data respectively. The cluster-based LM is 4-gram trained on TED using 500 classes.

After that, we experimented with two different DWL models. The first small DWL was trained on the TED corpus only. It improves the score on Test by 0.15 BLEU points while its effect on Dev is negligible. The second model is larger. It was trained on EPPS and NC in addition to TED. With the large DWL, the gain is much more important: 0.2 BLEU points on Dev and 0.4 BLEU points on Test. For our submission we used this last configuration.

System	Dev	Test
Baseline	40.17	34.12
With small DWL	40.19	34.27
With large DWL	40.40	**34.66**

Table 6: Experiments for English→French (MT)

8.3.1. SLT track

As a baseline for the SLT track, we used our best performing English→French MT system on the automatically punctuated and cased version of the SLT input. We experimented with different ways of tuning the SLT system. These experiments are shown in Table 7.

The baseline uses all the models mentioned in the previous section (Section 8.3) except the cluster-based LM and DWL. In this configuration, both Dev (Dev2010) and Test (Test2010) sets were automatically punctuated and cased with MCCI. We then translated the test set with a comparable MT system without retuning on the punctuated Dev. This MT system was also tuned on the Dev2010 (on its text version though) and to our surprise this outperforms the baseline by almost 0.7 BLEU points. We could even get an additional gain (more than 0.3 BLEU), by tuning on the same MT tuning set (Test2011). By translating the test set with our final MT system (adding the cluster-based LM and the DWL to the baseline), the performance of the system was boosted by an additional 0.7 BLEU points. This final system was used in our submission.

System	Dev	Test
Baseline	22.53	23.35
MT tuned	-	24.03
MT tuned (2011)	-	24.38
+ DWL + clusterLM	-	**25.05**

Table 7: Experiments for English→French (SLT)

8.4. English→Arabic

The raw data provided for this pair was processed similarly to our English→Arabic system last year [31]. We show the effect of the two main extensions for this year's submission in Table 8. The baseline's translation model is built by performing adaptation on two models. The first is trained on all parallel data (UN and TED) and the other is trained on TED only. It integrates a bilingual LM and a cluster-based LM (with 500 classes), and 4 more word-based LMs. Three of the word-based LMs were respectively trained on the provided corpora (TED, UN, and Giga), and the last one incorporates all Arabic data. We used the alternative pruning without retuning, which gave us a gain of 0.2 BLEU points. The transliteration of the untranslated words however has an unnoticeable effect (0.01). We decided to include it in our system since it is unlikely to hurt the system as it is applied only to untranslated words. The primary system we submitted applied the alternative pruning and the transliteration, while the contrastive one used our standard pruning and transliteration.

System	Dev	Test
Baseline	15.98	7.71
+ Pruning	-	7.91
+ Transliteration	-	**7.92**

Table 8: Experiments for English→Arabic (MT)

8.5. English→Chinese

This year we also participated in the text translation task of English→Chinese. There are four novel methods applied in this year's system. First we have applied the new MLT reordering model as described in Section 7. Secondly, we added the ECI corpus (LDC94T5) to train the language model. Thirdly we tuned the system with the data set Test2011 and tested it with Test2012. Last but not least we built the system based on Chinese words instead of on Chinese characters.

The system is trained on the bilingual TED and filtered UN corpora. Since the UN corpus is document-aligned, we performed sentence alignment using the Kuhn–Munkres (KM) algorithm [32]. For each sentence pair, we used the number of aligned word pairs which occur in a dictionary (corpus LDC2002L27) as the weight for the KM algorithm. We then set a threshold and selected the 30k best-matching sentences for training.

The language models are trained on the monolingual TED, ECI, Google n-grams and the target side of the whole UN data. The Chinese target side is segmented with the Stanford word segmenter[1].

Table 9 shows the improvements step by step. We report not only the BLEU score on the words ($Test_w$), but also the score on the Chinese characters ($Test_{char}$). Briefly, the reordering models and adaptation have given the main contribution to the improvement of translation quality. The baseline is a monotone translation with 6-gram language model. We have used the POS-based long-range reordering and the MLT reordering model in combination. The MLT reordering model yields a consistent improvement of about 0.3 BLEU points over the long-range reordering model. We use the TED corpus as the in-domain data to adapt the phrase table and language model. This adaptation on the TED corpus improves the results up to about 0.7 BLEU points. We have added three more language models besides the basic 6-gram one. *google1980LM* is a 5-gram language model trained on the Google n-grams of the 1980s. We have also tried to use all the Google n-grams. However, it does not help to use more data. *BiLM* is a 4-gram bilingual language model and *clusterLM* is a 4-gram cluster-based language model.

[1] http://nlp.stanford.edu/software/segmenter.shtml

System	Dev_w	$Test_w$	$Test_{char}$
Baseline	13.73	12.07	19.18
+ POS Reordering (long)	14.08	12.24	19.34
+ MLT Reordering	14.34	12.57	19.68
+ Adaptation	14.93	13.34	20.65
+ google1980LM	15.13	12.67	20.02
+ BiLM	15.20	12.95	20.32
+ clusterLM	15.18	**13.58**	**20.88**

Table 9: Experiments for English→Chinese (MT)

9. Conclusions

In this paper, we presented the systems with which we participated in the TED tasks of the IWSLT 2014 Evaluation Campaign. In total we submitted twelve systems for five language pairs, consisting of five primary MT systems, three contrastive ones, three primary SLT systems and one contrastive SLT system.

For all languages we used strong baseline systems, including various word and token-based language models, adaptation techniques and combinations of preordering and lexicalized reordering models. Careful data selection and inclusion of individual models trained on different data proved successful in many of the systems.

A new model this year is a reordering model that operates on multiple tree levels, which was applied successfully for English→Chinese.

Further improvements could be achieved for English→German by n-best list rescoring with language and translation models trained with Restricted Boltzmann Machines.

For translation into Arabic, a special phrase table pruning technique gave an improvement over the baseline. Even though the merits of a transliteration approach did hardly reflect in BLEU, they did not harm and helped to unify translation appearancein the Arabic target output.

We submitted contrastive systems in order to show the impact of our novel n-best list rescoring, adjective stemming and phrase extraction approaches for English→German, German→English and English→Arabic respectively.

A monolingual translation system for comma insertion and case correction played a vital role in adjusting the ASR output for speech translation and was successfully applied in all three SLT systems.

10. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

11. References

[1] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, 2003.

[2] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The KIT English-French Translation systems for IWSLT 2011," in *Proceedings of the 8th International Workshop on Spoken Language Translation*, San Francisco, CA, USA.

[3] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, 2003.

[4] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 2007.

[5] M. Mediani, J. Niehues, and A. Waibel, "Parallel Phrase Scoring for Extra-large Corpora," in *The Prague Bulletin of Mathematical Linguistics*, no. 98, 2012.

[6] J. Niehues and A. Waibel, "Detailed Analysis of Different Strategies for Phrase Table Adaptation in SMT," in *Proceedings of the 10th Conference of the Association for Machine Translation in the Americas*, San Diego, CA, USA, 2012.

[7] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[8] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *Proceedings of the International Conference on Spoken Language Processing*, Denver, CO, USA, 2002.

[9] K. Heafield, "KenLM: faster and smaller language model queries," in *Proceedings of the 6th Workshop on Statistical Machine Translation*, Edinburgh, United Kingdom, 2011.

[10] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Proceedings of the 6th Workshop on Statistical Machine Translation*, Edinburgh, United Kingdom, 2011.

[11] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes," in *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, Bergen, Norway, 1999.

[12] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, United Kingdom, 1994.

[13] A. N. Rafferty and C. D. Manning, "Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines," in *Proceedings of the Workshop on Parsing German*, 2008.

[14] D. Klein and C. D. Manning, "Accurate Unlexicalized Parsing," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, 2003.

[15] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, Skövde, Sweden, 2007.

[16] J. Niehues and M. Kolss, "A POS-Based Model for Long-Range Reorderings in SMT," in *Proceedings of the 4th Workshop on Statistical Machine Translation*, Athens, Greece, 2009.

[17] T. Herrmann, J. Niehues, and A. Waibel, "Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation," in *Proceedings of the 7th Workshop on Syntax, Semantics and Structure in Statistical Translation*, Altanta, GA, USA, 2013.

[18] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation," in *Proceedings of the 2nd International Workshop on Spoken Language Translation*, Pittsburgh, PA, USA, 2005.

[19] J. Niehues and A. Waibel, "An MT Error-Driven Discriminative Word Lexicon using Sentence Structure Features," in *Proceedings of the 8th Workshop on Statistical Machine Translation*, Sofia, Bulgaria, 2013.

[20] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Ann Arbor, MI, USA, 2005.

[21] E. Cho, J. Niehues, and A. Waibel, "Segmentation and Punctuation Prediction in Speech Language Translation using a Monolingual Translation System," in *Proceedings of the 9th International Workshop on Spoken Language Translation*, Hong Kong, 2012.

[22] J. Niehues and A. Waibel, "Continuous Space Language Models using Restricted Boltzmann Machines," in *Proceedings of the 9th International Workshop on Spoken Language Translation*, Hong Kong, 2012.

[23] J. Niehues, A. Allauzen, F. Yvon, and A. Waibel, "Combining Techniques from Different NN-based Language Models for Machine Translation," in *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas*, Vancouver, BC, Canada, 2014.

[24] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and Robust Neural Network Joint Models for Statistical Machine Translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, MD, USA, 2014.

[25] Z. Cao, T. Qin, T. yan Liu, M.-F. Tsai, and H. Li, "Learning to Rank: From Pairwise Approach to Listwise Approach," in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, USA, 2007.

[26] E. Deza and M. Deza, *Dictionary of Distances*. North-Holland, 2006.

[27] P. Nakov and J. Tiedemann, "Combining Word-level and Character-level Models for Machine Translation Between Closely-related Languages," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Korea, 2012.

[28] G. Wu, Y. Zhang, and A. Waibel, "Rule-Based Preordering on Multiple Syntactic Levels in Statistical Machine Translation," in *Proceedings of the 11th International Workshop on Spoken Language Translation*, Lake Tahoe, USA, 2014.

[29] R. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010.

[30] H. Schmid and F. Laws, "Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-Grained POS Tagging," in *Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, United Kingdom, 2008.

[31] T.-L. Ha, T. Herrmann, J. Niehues, M. Mediani, E. Cho, Y. Zhang, I. Slawik, and A. Waibel, "The KIT Systems for IWSLT 2013," in *Proceedings of the 10th International Workshop on Spoken Language Translation*, Heidelberg, Germany, 2013.

[32] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, 1955.

NTT-NAIST Syntax-based SMT Systems for IWSLT 2014

Katsuhito Sudoh[*], *Graham Neubig*[†], *Kevin Duh*[†], *Katsuhiko Hayashi*[*]

[*]NTT Communication Science Laboratories, Seika-cho, Kyoto, Japan
[†]Nara Institute of Science and Technology (NAIST), Ikoma-shi, Nara, Japan
sudoh.katsuhito@lab.ntt.co.jp

Abstract

This paper presents NTT-NAIST SMT systems for English-German and German-English MT tasks of the IWSLT 2014 evaluation campaign. The systems are based on generalized minimum Bayes risk system combination of three SMT systems using the forest-to-string, syntactic pre-ordering, and phrase-based translation formalisms. Individual systems employ training data selection for domain adaptation, truecasing, compound word splitting (for German-English), interpolated n-gram language models, and hypotheses rescoring using recurrent neural network language models.

1. Introduction

Spoken language is a very important and also challenging target for machine translation (MT). MT tasks in the IWSLT evaluation campaign focus on the translation of TED Talks subtitles. These subtitles tend to be clean transcriptions with few disfluencies, and the talks themselves are logically and syntactically well-organized compared to casual conversations.

In order to take advantage of this fact, our system this year use syntax-based statistical machine translation (SMT) techniques, which allow for the use of source-side syntactic knowledge to improve translation accuracy. Specifically, we use forest-to-string (F2S) translation and syntax-based pre-ordering. The overall system was based on a combination of three systems based on F2S, pre-ordering, and standard PBMT, and includes domain adaptation of translation and language models, rescoring using neural network language models, and compound splitting for German.

Specifically comparing to our system from last year's competition [1], we have made two improvements. The first is that we tested a new hypergraph search algorithm [2] in the F2S system, and compare it to the more traditional method of cube pruning. The second is that this year we attempted to extract pre-ordering rules automatically from parallel corpora, as opposed to hand-designing preordering rules based on linguistic intuition.

This paper presents details of our systems and reports the official results together with some detailed discussions on contributions of the techniques involved.

2. Individual Translation Methods

We use three different translation methods and combine the results through system combination. Each of the three methods is described in this section, focusing especially on our new attempts this year on forest-to-string and pre-ordering.

2.1. Forest-to-String Machine Translation

In our previous year's submission to IWSLT, we achieved promising results using the forest-to-string machine translation (F2S; [3]) framework. F2S is a generalization of tree-to-string machine translation (T2S; [4]) that performs translation by first syntactically parsing the source sentence, then translating from sub-structures of a packed forest of potential parses to a string in the target language.

We have previously found that F2S produces highly competitive results for language pairs with large divergence in syntax such as Japanese-English or Japanese-Chinese [5]. However, we have also found that there are several elements that must be appropriately handled to achieve high translation accuracy using syntax-driven methods [6], one of which is search. In the F2S component of our submission to IWSLT this year, we experimented with two different search algorithms to measure the effect that search has on the German-English and English-German pairs.

As the first algorithm, we use the standard method for search in tree-based methods of translation: *cube pruning* [7]. For each edge to be expanded, cube pruning sorts the child hypotheses in descending order of probability, and at every step pops the highest-scoring hypothesis off the stack, calculates its language model scores, and adds the popped, scored edge to the hypergraph. It should be noted that the LM scores are not calculated until after the edge is popped, and thus the order of visiting edges is based on only an LM-free approximation of the true edge score, resulting in search errors.

In our F2S system this year, we test a new method of *hypergraph search* [2], which aims to achieve better search accuracy by considering the characteristics of LM states when deciding the order in which to calculate edges. Particularly, it exploits the fact that states with identical unigram contexts are likely to have similar probabilities, and groups these together at the beginning of the search. It then proceeds to split these states into bi-gram or higher order contexts gradu-

Proceedings of the 11th International Workshop on Spoken Language Translation
Lake Tahoe, December 4th and 5th, 2014

ally, refining the probability estimates until the limit on number of stack pops is reached. In our previous work [6] we have found that hypergraph search achieved superior results to cube pruning, and we hypothesize that these results will carry over to German-English and English-German as well.

2.2. Syntax-based Pre-ordering

Pre-ordering is a method that attempts to first reorder the source sentence into a word order that is closer to the target, then translate using a standard method such as PBMT. We used hand-crafted German-English pre-ordering rules [8] in our submission last year. This year's system uses an automatic method to extract domain-dependent pre-ordering rules, avoiding the time-consuming effort required for creating hand-crafted rules. The pre-ordering method is basically similar to [9], but is limited to reordering of child nodes in syntactic parse trees rather than rewriting and word insertion.

Since the pre-ordering does not work perfectly in all cases, we allow for further reordering in the PBMT system that translates the preordered sentences. The reordering limit of this system is chosen experimentally using held-out data (dev. set BLEU in this paper).

2.2.1. Reordering Pattern Extraction

A reordering pattern represents a reordering of child nodes in a source language parse tree, determined by word alignment. The reordering pattern is similar to a tree-based translation pattern called *frontier graph fragments*, which form the most basic unit in tree-based translation [10], but only holds reordering information on the non-terminal child nodes. A reordering pattern can be extracted from an *admissible* node [11] in the parse tree that covers a distinct contiguous spans in the corresponding target language sentences. Since such a reordering pattern only is constrained by the syntactic labels on the parent and child nodes, we consider several attributes of reordering patterns: syntactic labels of its grand-parent, left and right siblings of the parent, and surface forms of its child nodes (only when the child is a part-of-speech node).

2.2.2. Deterministic Pre-ordering

In order to make the pre-ordering deterministic, we use reordering rules from dominant reordering patterns that agree with more than 75% on the same source language subtrees. Here, additional attributes define more specific rules that are not applied to the subtrees with different attributes.

We apply these reordering rules greedily to the syntactic parse tree in descending order of preference from specific (more attributes) to general (less attributes) rules. If different rules with the same number of attributes can be applied, the most probable one is chosen. More details about the method can be found in [9].

2.3. Standard Phrase-based Translation

Phrase-based machine translation (PBMT; [12]) models the translation process by splitting the source sentence into phrases, translating the phrases into target phrases, and reordering the phrases into the target language order. PBMT is currently the most widely used method in SMT as it is robust, does not require the availability of linguistic analysis tools, and achieves high accuracy, particularly for languages with similar syntactic structure.

3. Additional System Enhancements

Here we review techniques that were used in our submission last year [1] and also describe some of our new attempts that were not effective in our pilot test and not included in the final system.

3.1. Training Data Selection

The target TED domain is different in both style and vocabulary from many of the other bitexts, e.g. Europarl, CommonCrawl (which we collectively call "general-domain" data). To address this domain adaption problem, we performed adaptation training data selection using the method of [13].[1] The intuition is to select general-domain sentences that are similar to in-domain text, while being dis-similar to the average general-domain text.

To do so, one defines the score of an general-domain sentence pair (e, f) as [14]:

$$[IN_E(e) - GEN_E(e)] + [IN_F(f) - GEN_F(f)] \quad (1)$$

where $IN_E(e)$ is the *length-normalized* cross-entropy of e on the English in-domain LM. $GEN_E(e)$ is the length-normalized cross-entropy of e on the English general-domain LM, which is built from a sub-sample of the general-domain text. Similarly, $IN_F(f)$ and $GEN_F(f)$ are the cross-entropies of f on Foreign-side LM. Finally, sentence pairs are ranked according to Eq. 1 and those with scores lower than some empirically-chosen threshold are added together with the in-domain bitext for translation model training. Here, the LMs are Recurrent Neural Network Language Models (RNNLMs), which have been shown to outperform n-gram LMs in this problem [13].

3.2. German Compound Word Splitting

German compound words present sparsity challenges for machine translation. To address this, we split German words following the general approach of [15]. The idea is to split a word if the geometric average of its subword frequencies is larger than whole word frequency. In our implementation, for each word, we searched for all possible decompositions into two sub-words, considering the possibility of deleting common German fillers "e", "es", and "s" (as in "Arbeit+s+tier"). The unigram frequencies for the subwords and

[1] Code/scripts available at http://cl.naist.jp/∼kevinduh/a/acl2013

whole word is computed from the German part of the bitext. This simple algorithm is especially useful for handling out-of-vocabulary and rare compound words that have high frequency sub-words in the training data. For the F2S system, sub-words are given the same POS tag as the original whole word.

In the evaluation campaign, we performed compound splitting only in the German-to-English task. We do not attempt to split German words for the English-to-German task, since it is non-trivial to handle recombination of German split words after reordering and translation.

3.3. RNNLM Rescoring

Continuous-space language models using neural networks have attracted recent attention as a method to improve the fluency of output of MT or speech recognition. In our system, we used the recurrent neural network language model (RNNLM) of [16].[2] This model uses a continuous space representation over the language model state that is remembered throughout the entire sentence, and thus has the potential to ensure the global coherence of the sentence to the greater extent than simpler n-gram language models.

We incorporate the RNNLM probabilities through rescoring. For each system, we first output a 10,000-best list, then calculate the RNNLM log probabilities and add them as an additional feature to each translation hypothesis. We then re-run a single MERT optimization to find ideal weights for this new feature, and then extract the 1-best result from the 10,000-best list for the test set according to these new weights. The parameters for RNNLM training are tuned on the dev set to maximize perplexity, resulting in 300 nodes in the hidden layer, 300 classes, and 4 steps of back-propagation through time.

3.4. GMBR System Combination

We used a system combination method based on Generalized Minimum Bayes Risk optimization [17], which has been successfully applied to different types of SMT systems for patent translation [18]. Note that our system combination only picks one hypothesis from an N-best list and does not generate a new hypothesis by mixing partial hypotheses among the N-best.

3.4.1. Theory

Minimum Bayes Risk (MBR) is a decision rule to choose hypotheses that minimize the expected loss. In the task of SMT from a French sentence (f) to an English sentence (e), the MBR decision rule on $\delta(f) \rightarrow e'$ with the loss function L over the possible space of sentence pairs ($p(e, f)$) is denoted as:

$$\underset{\delta(f)}{\operatorname{argmin}} \sum_{e} L(\delta(f)|e)p(e|f) \tag{2}$$

[2] http://www.fit.vutbr.cz/~imikolov/rnnlm/

In practice, we approximate this using N-best list $N(f)$ for the input f.

$$\underset{e' \in N(f)}{\operatorname{argmin}} \sum_{e \in N(f)} L(e'|e)p(e|f) \tag{3}$$

Although MBR works effectively for re-ranking single system hypotheses, it is challenging for system combination because the estimated $p(e|f)$ from different systems cannot be reliably compared. One practical solution is to use uniform $p(e|f)$ but this does not achieve Bayes Risk. GMBR corrects by parameterizing the loss function as a linear combination of sub-components using parameter $\boldsymbol{\theta}$:

$$L(e'|e;\boldsymbol{\theta}) = \sum_{k=1}^{K} \theta_k L_k(e'|e) \tag{4}$$

For example, suppose the desired loss function is "1.0−BLEU". Then the sub-components could be "1.0−precision(n-gram) $(1 \leq n \leq 4)$" and "brevity penalty".

Assuming uniform $p(e|f)$, the MBR decision rule can be denoted as:

$$\begin{aligned} &\underset{e' \in N(f)}{\operatorname{argmin}} \sum_{e \in N(f)} L(e'|e;\boldsymbol{\theta}) \frac{1}{|N(f)|} \\ &= \underset{e' \in N(f)}{\operatorname{argmin}} \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e'|e) \end{aligned} \tag{5}$$

To ensure that the uniform hypotheses space gives the same decision as the original loss in the true space $p(e|f)$, we use a small development set to tune the parameter $\boldsymbol{\theta}$ as follows. For any two hypotheses e_1, e_2, and a reference translation e_r (possibly not in $N(f)$) we first compute the true loss: $L(e_1|e_r)$ and $L(e_2|e_r)$. If $L(e_1|e_r) < L(e_2|e_r)$, then we would want $\boldsymbol{\theta}$ such that:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) < \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \tag{6}$$

so that GMBR would select the hypothesis achieving lower loss. Conversely if e_2 is a better hypothesis, then we want opposite relation:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) > \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \tag{7}$$

Thus, we directly compute the true loss using a development set and ensure that our GMBR decision rule minimizes this loss.

3.4.2. Implementation

We implement GMBR for SMT system combination as follows.

First we run SMT decoders to obtain N-best lists for all sentences in the development set, and extract all pairs of hypotheses where a difference exists in the true loss. Then we optimize θ in a formulation similar to a Ranking SVM [19]. The pair-wise nature of Eqs. 6 and 7 makes the problem amendable to solutions in "learning to rank" literature [20]. We used BLEU as the objective function and the subcomponents of BLEU as features (system identity feature was not used). There is one regularization hyperparameter for the Ranking SVM, which we set by cross-validation over the development set (`dev2010`).

3.5. What Didn't Work Immediately

This year we tried to include a state-of-the-art Neural Network Joint Model (NNJM) [21] to improve the accuracy of translation probability estimation. The model is used to predict a target language word using its three preceding target language words and eleven source language words surrounding its affiliation (the non-NULL source language word aligned to the target language word to be predicted). We used top 16,000 source and target vocabularies in the model and mapped the other words into a single OOV symbol, while the original paper[21] used part-of-speech classes. Although the original paper presented a method for integrating the model with decoding, we used the NNJM for reranking n-best hypotheses in a similar manner as the RNNLM described above. The NNJM gave some improvements from the baseline 1-best in our pilot test, but they were much smaller than those resulting from RNNLM, and when the NNJM was combined with RNNLM we saw no significant gains. One possible reason is the small training data size; the model is very sparse and needs large training data because of its large contexts of fourteen (eleven source and three target) words. The affiliation is very important to predict the target word correctly but it was determined by automatic word alignment (such as GIZA++) and may not always be good enough in our experiments.

We also tried *post-ordering* [22] by shift-reduce reordering [23] for German-to-English. It was not effective in our pilot test even in the first-pass lexical translation, probably due to less effective English-to-German pre-ordering rules.

4. Experiments

We conducted experiments on the English-German and German-English MT tasks using the SMT systems described above developed using the supplied datasets.

4.1. Setup

4.1.1. System Overview

We used three individual SMT systems presented in Section 2: forest-to-string (F2S), phrase-based with pre-ordering (Preorder), and phrase-based without pre-ordering (PBMT). F2S was implemented with Travatar[3] [24] and the phrase-based MT systems were implemented with Moses [25].

For the Travatar rule tables, we used a modified version of Egret[4] as a syntactic parser, and created forests using dynamic pruning including all edges that occurred in the 100-best hypotheses. We trained the parsing model using the Berkeley parser over the Wall Street Journal section of the Penn Treebank[5] for English, and TIGER corpus [26] for German. For model training, the default settings for Travatar were used, with the exception of changing the number of composed rules to 6 with Kneser-Ney smoothing. For search in the F2S models, we used the previously described hypergraph search method.

For the Moses phrase tables, we used standard training settings with Kneser-Ney smoothing of phrase translation probabilities [27].

4.1.2. Translation Models

We trained the translation models using WIT[3] training data (178,526 sentences) and 1,000,000 sentences selected over other bitexts (Europarl, News Commentary, and Common Crawl) by the method described in Section 3.1.

4.1.3. Language Models

We used word 5-gram language models of German and English that were linearly interpolated from several word 5-gram language models trained on different data sources (WIT[3], Europarl, News Commentary, and Common Crawl). The interpolation weights were optimized to minimize perplexity on the development set, using `interpolate-lm.perl` in Moses. Individual language models were trained by SRILM with modified Kneser-Ney smoothing.

4.1.4. Truecaser

In order to maintain the casing of words across languages, we opted to use truecasing (based on the Moses truecaser) on both the source and target sides. Truecasing keeps the case of all words that are not sentence initial, and chooses the case of the sentence initial word based on the most frequent appearance among different cases in the training data.

4.2. Full System Results

Our full system was a GMBR-based combination of F2S, Preorder, and PBMT. Tables 1 and 2 show the official evaluation results for English-to-German and German-to-English tasks, respectively. Among the individual systems, F2S showed the best BLEU and TER, and Preorder was the worst. The poor performance of Preorder was not consistent with our development results on older test sets (discussed later)

[3] `http://www.phontron.com/travatar/`
[4] `https://github.com/neubig/egret/`
[5] `http://www.cis.upenn.edu/~treebank/`

Table 1: Official results for English-to-German (case sensitive). Δ*BestSingle* represents the differences from the results by the best single system (F2S).

System	tst2013		tst2014	
(En-De)	BLEU	TER	BLEU	TER
Combination	.2580	.5386	.2209	.5760
Δ*BestSingle*	+.0097	-.0103	-.0021	-.0100
F2S	.2483	.5489	.2230	.5860
Preorder	.2443	.5567	.2112	.5947
PBMT	.2453	.5528	.2150	.5906

Table 2: Official results for German-to-English (case sensitive).

System	tst2013		tst2014	
(De-En)	BLEU	TER	BLEU	TER
Combination	.2781	.5162	.2377	.5643
Δ*BestSingle*	+.0070	-.0224	+.0030	-.0180
F2S	.2711	.5386	.2347	.5823
Preorder	.2646	.5425	.2208	.5914
PBMT	.2671	.5422	.2229	.5885

Table 3: Percentages of individual system outputs chosen by system combination.

System	En-De		De-En	
	tst2013	tst2014	tst2013	tst2014
F2S	16.11	19.16	57.59	54.24
Preorder	49.14	50.34	39.69	42.72
PBMT	34.74	30.50	1.39	3.04

and our last year's results with hand-crafted rules [1]. The GMBR combination further improved BLEU and TER compared to those of F2S, except for BLEU in tst2014. The improvement in TER was large, about 1% in English-to-German and 2% in German-to-English, compared to an at most 1% gain in BLEU.

Table 3 shows the contributions of individual systems in the system combination, by percentages of *chosen* system outputs. As we discussed in our system description paper last year [1], the GMBR system combination works as voting over n-best hypotheses from different systems. The results in Table 3 indicate the best F2S system contributed little in English-German and the worst Preorder system contributed about a half of the system combination outputs. There were large difference between these results and our last year's results, but we do not yet have a solid answer for the reason. One possibility is the inconsistency between the training condition (Preorder worked well) and the test condition (Preorder worked poorly) as discussed later in detail.

Table 4: Results on old IWSLT test sets for English-to-German (case sensitive). Scores in **bold** indicate the best individual system results.

System	tst2010		tst2011		tst2012	
(En-De)	BLEU	TER	BLEU	TER	BLEU	TER
Combi.	.2516	.6309	.2714	.5870	.2388	.6380
F2S	**.2487**	**.6452**	**.2670**	**.5989**	**.2306**	**.6545**
Preorder	.2412	.6523	.2639	.6043	.2274	.6601
PBMT	.2419	.6509	.2634	.6031	.2280	.6575

Table 5: Results on old IWSLT test sets for German-to-English (case sensitive). Scores in **bold** indicate the best individual system results.

System	tst2010		tst2011		tst2012	
(De-En)	BLEU	TER	BLEU	TER	BLEU	TER
Combi.	.3155	.5583	.3711	.4949	.3144	.5515
F2S	.3037	.5901	.3465	.5313	.3028	.5812
Preorder	**.3065**	**.5730**	**.3604**	**.5088**	**.3055**	**.5647**
PBMT	.3043	.5754	.3571	.5119	.3038	.5678

4.3. Detailed Results and Discussions

4.3.1. Evaluation on Old Test Sets

Tables 4 and 5 shows the results on old IWSLT test sets (tst2010, tst2011, tst2012). The results tend to show a different trend than those for tst2013 and tst2014; Specifically looking at the German-to-English task, F2S was the worst and Preorder worked the best on these older data sets, as shown in Table 5.

One possible reason for this difference is the difference in the *original* languages in the older and newer test sets. The official test sets this year (tst2013, tst2014) came from TEDX talks in German, and thus the source German sentences were transcriptions. In contrast, the older test sets (tst2010, tst2011, tst2012) came from TED talks in English, and thus the source German sentences were translations from English. It has been widely noted that translations differ significantly from original texts stylistically (e.g. [28]), and the difference may cause some inconsistencies in syntactic parsing and syntax-based translation. Preorder used only dominant reordering patterns in German extracted from translated German sentences, which were consistent with the TED test sets but not with the TEDX test sets.

4.3.2. Effect of Search on F2S Translation

As mentioned in Section 2.1, we tested two algorithms for search in F2S models, cube pruning, and hypergraph search. In Figure 1 we show the speed and accuracy for both algorithms at various beam sizes for English-German and German-English translation. All results are reported on tst2010, but similar results were found for other sets.

From these results, we can see that given an identical de-

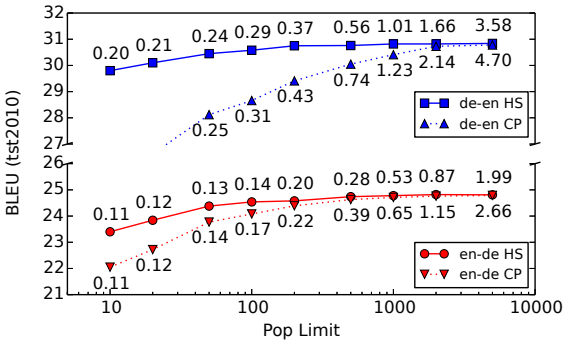


Figure 1: Hypergraph search (HS) and cube pruning (CP) results for F2S translation. Numbers above and below the lines indicate time in seconds/sentence for HS and CP respectively.

coding time, hypergraph search outperforms cube pruning on both language pairs at all beam sizes, especially for smaller beams. This effect was particularly notable for German-English translation. Even when the beam is reduced from 5000 (which was used in our actual submission) to 10, we only see a drop in one BLEU point, but reduce the time required for decoding to 200ms, much of which can be attributed to processing other than search such as rule lookup or file input/output. This is in contrast to cube pruning, which sees a 5.5 BLEU point drop at the same beam size.

5. Conclusion

In this paper, we presented our English-to-German and German-to-English SMT systems using combination of forest-based, pre-ordering, and standard phrase-based MT systems. The forest-based system employed the hypergraph search for efficient translation, and the pre-ordering used automatically-induced rules from the bilingual corpus. The individual systems used training data selection, compound word splitting for German, and RNNLM rescoring, same as our last year's systems. Our results show the forest-to-string SMT was consistently the most effective of the three and can be further improved by GMBR system combination with the results from the other two systems. The pre-ordering was not effective in the 2013 and 2014 test sets in contrast to the older ones.

6. Acknowledgements

We would like to thank the anonymous reviewer for the comments to improve this system description paper. We also appreciate Dr. Jun Suzuki for his support for the use of NNJM in our experiments.

7. References

[1] K. Sudoh, G. Neubig, K. Duh, and H. Tsukada, "NTT-NAIST SMT Systems for IWSLT 2013," in *Proceedings of the 10th International Workshop on Spoken Language Translation (IWSLT 2013)*, Heidelberg, Germany, December 2013.

[2] K. Heafield, P. Koehn, and A. Lavie, "Grouping language model boundary words to speed k–best extraction from hypergraphs," in *Proceedings of the 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 958–968.

[3] H. Mi, L. Huang, and Q. Liu, "Forest-based translation," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008, pp. 192–199.

[4] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation," in *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2006.

[5] G. Neubig, "Forest-to-string SMT for asian language translation: NAIST at WAT2014," in *Proceedings of the 1st Workshop on Asian Translation (WMT)*, 2014.

[6] G. Neubig and K. Duh, "On the elements of an accurate tree-to-string machine translation system," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

[7] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[8] M. Collins, P. Koehn, and I. Kucerova, "Clause Restructuring for Statistical Machine Translation," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, June 2005, pp. 531–540.

[9] F. Xia and M. McCord, "Improving a Statistical MT System with Automatically Learned Rewrite Patterns," in *Proceedings of Coling 2004*, Geneva, Switzerland, August 2004, pp. 508–514.

[10] M. Galley, M. Hopkins, K. Knight, and D. Marcu, "What's in a translation rule?" in *HLT-NAACL 2004: Main Proceedings*, Boston, Massachusetts, USA, May 2004, pp. 273–280.

[11] W. Wang, K. Knight, and D. Marcu, "Binarizing Syntax Trees to Improve Syntax-Based Machine Translation Accuracy," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, June 2007, pp. 746–754.

[12] P. Koehn, F. J. Och, and D. Marcu, "Statistical Phrase-Based Translation," in *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Canada, May 2003, pp. 48–54.

[13] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation Data Selection using Neural Language Models: Experiments in Machine Translation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria, August 2013, pp. 678–683.

[14] A. Axelrod, X. He, and J. Gao, "Domain Adaptation via Pseudo In-Domain Data Selection," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., July 2011, pp. 355–362.

[15] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, 2003, pp. 187–194.

[16] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, Chiba, Japan, 2010, pp. 1045–1048.

[17] K. Duh, K. Sudoh, X. Wu, H. Tsukada, and M. Nagata, "Generalized Minimum Bayes Risk System Combination," in *Proceedings of 5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand, November 2011, pp. 1356–1360.

[18] K. Sudoh, K. Duh, H. Tsukada, M. Nagata, X. Wu, T. Matsuzaki, and J. Tsujii, "NTT-UT Statistical Machine Translation in NTCIR-9 PatentMT," in *Proceedings of the 9th NTCIR Conference*, Tokyo, Japan, December 2011.

[19] T. Joachims, "Training Linear SVMs in Linear Time," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 217–226.

[20] C. He, C. Wang, Y.-X. Zhong, and R.-F. Li, ""a survey on learning to rank"," in *Proceedings on 2008 International Conference on Machine Learning and Cybernetics*, vol. 3, 2008, pp. 1734–1739.

[21] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and Robust Neural Network Joint Models for Statistical Machine Translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 1370–1380. [Online]. Available: http://www.aclweb.org/anthology/P14-1129

[22] K. Sudoh, X. Wu, K. Duh, H. Tsukada, and M. Nagata, "Post-ordering in statistical machine translation," in *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, Xiamen, China, September 2011, pp. 316–323.

[23] K. Hayashi, K. Sudoh, H. Tsukada, J. Suzuki, and M. Nagata, "Shift-Reduce Word Reordering for Machine Translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, October 2013, pp. 1382–1386.

[24] G. Neubig, "Travatar: A Forest-to-String Machine Translation Engine based on Tree Transducers," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Sofia, Bulgaria, August 2013, pp. 91–96.

[25] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic, June 2007, pp. 177–180.

[26] S. Brants, S. Dipper, P. Eisenberg, S. Hansen-Schirra, E. h. König, W. Lezius, C. Rohrer, G. Smith, and H. Uszkoreit, "TIGER: Linguistic interpretation of a German corpus," *Research on Language and Computation*, vol. 2, no. 4, pp. 597–620, 2004.

[27] G. Foster, R. Kuhn, and H. Johnson, "Phrasetable Smoothing for Statistical Machine Translation," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, July 2006, pp. 53–61.

[28] M. Koppel and N. Ordan, "Translationese and its dialects," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011, pp. 1318–1326.

The USTC Machine Translation System for IWSLT 2014

Shijin Wang[+*], *Yuguang Wang*[*], *Jianfeng Li*[*], *Yiming Cui*[*], *Lirong Dai*[+]

[+]National Engineering Laboratory of Speech and Language Information Processing,
University of Science and Technology of China
[*]IFLYTEK Co. LTD.
shijinwang@ustc.edu
{ygwang2, jfli3, ymcui}@iflytek.com
lrdai@ustc.edu.cn

Abstract

This paper describes the University of Science and Technology of China's (USTC) system for the MT track of IWSLT2014 Evaluation Campaign. We participated in the Chinese-English and English-Chinese translation tasks. For both tasks, we used a phrase-based statistical machine translation system (SMT) as our baseline. To improve the translation performance, we applied a number of techniques, such as word alignment with the l_0-norm, phrase table smoothing, hierarchical reordering model, domain adaptation of the language and translation model, recurrent neural network based language model, neural network joint model, etc. By integrating these techniques, we obtained total improvements of 4.2% BLEU score for Chinese-English system and 3.7% BLEU score for English-Chinese system, compared to the baseline systems.

1. Introduction

In the IWSLT 2014 evaluation campaign, we participated in the optional MT track with the Chinese-English and English-Chinese translation tasks. We build a phrase-based statistical machine translation system for these tasks, and similar techniques are applied to Chinese-English and English-Chinese systems.

Before training, Chinese sentences are segmented into words using our Chinese word segmentation tool, and English sentences are tokenized and transformed into lower case. After preprocessing, GIZA++ is applied for training word alignments. Then, bilingual phrase pairs are extracted from word aligned parallel sentences. Based on the extracted phrase table, we build a weak baseline system with several widely-used features. The feature weights are tuned using Minimum Error Rate Training (MERT) [1].

By refining some steps in the training process we obtained our strong baseline. Firstly, we tried different development set. Secondly, we modified GIZA++ with the l_0-norm [2]. Then we tried different heuristics to combine bidirectional word alignment results. When calculating the phrase translation probabilities, we adopted Good-Turing smoothing rather than using relative frequency. By also using hierarchical reordering model (HRM) and k-best Margin Infused Relaxed Algorithm (kbMIRA) [3], our strong baseline system obtained significant improvements over the weak baseline.

To further improve translation performance, we exploited additional models, including more and larger language models, neural network based models, out-of-domain models trained from MultiUN corpus, and an operation sequence model [4]. We use these models in two ways: one is to integrate them into the decoder, and the other is to use them to rerank the n-best translations generated by the decoder.

Language models play an important role in our statistical machine translation system. Besides the in-domain language model trained from the TED training corpus, we built several larger language models from English Gigaword corpus and News Crawl corpora provided by the evaluation campaign. These language models were added into the translation system as separate features. We also built a word class based language model to alleviate data sparseness. Furthermore, a backward language model is used in reranking.

Neural networks have been successfully applied to machine translation recently. In our system, we built a recurrent neural network language model (RNNLM) for reranking. We also built several neural network joint models (NNJM), one for decoding, and the others for reranking.

The rest of the paper is organized as follows. In section 2, we generally describe the techniques we adopted in the translation systems. In section 3, we illustrate our experimental results on Chinese-English and English-Chinese translation systems. In the last section, we give a brief conclusion and the future work.

2. System Overview

For the IWSLT 2014 evaluation campaign, we build a phrase-based statistical machine translation system that is based on a log-linear discriminative model.

2.1. SMT System

Our phrase-based statistical machine translation system is mainly based on the work of an open-source toolkit Moses [5]. A number of widely used features are adopted in our SMT system, including bidirectional phrase translation probabilities and lexical translation probabilities, language model, word penalty, phrase penalty, distance-based distortion model, and hierarchical reordering model [6].

We use a modified GIZA++ toolkit for word alignment, which extend the IBM models and HMM model by the addition of an l_0 prior to the word-to-word translation model. It can reduce overfitting, and generate less useless phrase pairs. We test different heuristics (*grow*, *grow-diag-final*, *grow-diag-final-and*) for symmetrizing bidirectional word alignment results. For different tasks, there are some notable differences in performance among heuristics. When calculating the phrase translation probabilities, we use Good-Turing smoothing techniques, rather than using relative frequency. It turned out to be useful to improve translation performance.

Since the SMT system is based on a log-linear model, feature weights have a big impact on translation quality.

While tuning feature weights, we tried different development sets. In addition, tuning algorithm also makes some difference. We tested MERT and kbMIRA, and found that kbMIRA is better than MERT in our experiments.

N-gram language models are created with the SRILM toolkit [7]. We evaluate the tokenized translation results in case-sensitive fashion, using the BLEU metric [8].

For date, time and other number related expressions (DTN), we have some special treatments. We firstly write some rules to identify DTN expressions in source language, and then edit corresponding translations in target language for each identification rule. Regular expressions are used for the task. Finally, these rules with translations are added into the translation model with high translation probabilities.

Some source words, which cannot be translated by the translation model, are called out-of-vocabulary (OOV) words. We make additional process for two kinds of OOV words. The first case is those do occur in the TED training corpus, but no corresponding translations in the phrase table due to the restriction of phrase extraction. In this case, we make use of lexical translation table to translate these OOV words. The second case is those do not occur in TED corpus but appear in MultiUN corpus. For these words, we extract their translation from the MultiUN phrase table. In the other cases, we simply drop OOV words.

To exploit some features that are not suitable to be added into the decoder, we use them in the reranking step. The n-best translation results are generated by the decoder, and then additional feature scores are calculated for each hypothesis. Finally, the n-best list is reranked according to the new feature set.

Along with the techniques mentioned above, we also implement some novel models to further improve translation performance, which are described as follows.

2.2. Language Model

We put an emphasis on language modeling. Besides the 5-gram model trained from TED corpus, we also train several n-gram language models from the English Gigaword corpus and News Crawl corpora. Each of them is taken as a separate feature in the log-linear model. In addition, we build several other types of language models described below.

2.2.1. Backward Language Model

We build a backward n-gram language model [9], where the probability of each word is estimated depending on words following it:

$$P(W) = \sum_{i=T}^{1} P(w_i \mid w_{i+1}, w_{i+2}, ..., w_{i+n-1}) \quad (1)$$

We use the model in reranking stage. In our experiments, the backward language model can sometimes be helpful, but not always.

2.2.2. Class-based Language Model

Data sparseness is a common problem in natural language processing. Automatically clustering words from monolingual or bilingual training corpora into word classes is a widely used method to improving statistical models [10]. Here we build a class-based language model, and find it helpful in improving translation quality.

Firstly, we made use of `mkcls` in Moses toolkit to train a mapping from each word to a fixed class. Then we project words in training corpus to classes and train a class-based language model. In our system, a 7-gram class-based model is trained using SRILM toolkit. Class-based language model probability is used as a separate feature in decoder.

2.2.3. Recurrent Neural Network Language Model

Recent work has shown that recurrent neural network language models outperform significantly the n-gram models, even in case when n-gram models are trained on much more data. Moreover, when compared to feed-forward neural network language model, the RNNLM allows effective processing of sequences and patterns with arbitrary length, and it enables to learn long-distance dependence in the hidden layer.

In our system, we use the open-source RNNLM toolkit [11] to train a recurrent neural network language model. The model is used at the reranking stage to generate an additional feature for each hypothesis.

2.3. Neural Network Joint Model

Neural network based technologies are playing a more and more important role in recent natural language processing research. Recent studies on machine translation, which introduce neural network language model (NNLM) as features, turns out to be a breakthrough progress [12]. Moreover, some researchers present a novel formulation of a neural network joint model (NNJM) [13] as an extension of NNLM, which introduces dependence on source words. Though NNJM is just based on a lexicalized probabilistic model and a simple feed forward neural network, the experimental results show that it has significant improvements over the baseline systems.

The basic NNJM (s2t.l2r) formula can be written as:

$$P(T \mid S) \approx \prod_{i=1}^{|T|} P(t_i \mid t_{i-1}, ..., t_{i-n+1}, \xi_i) \quad (2)$$

where T is the target sentence, S is the source sentence, ξ_i is the source word window. In this circumstance, each target word t_i is affiliated with exactly one source word at index a_i. Then ξ_i is a m-word source window centered at a_i.

$$\xi_i = s_{a_i-(m-1)/2}, ..., s_{a_i}, ..., s_{a_i+(m-1)/2} \quad (3)$$

By changing the dependence order among target words, or swapping source and target languages, we can implement several variants of NNJM (s2t.r2l, t2s.l2r, t2s.r2l) as shown in Equation 4 to 6, where ζ_i is similar with ξ_i, which is just a replacement of source word s into target word t.

$$P(T \mid S) \approx \prod_{i=1}^{|T|} P(t_i \mid t_{i+1}, ..., t_{i+n-1}, \xi_i) \quad (4)$$

$$P(S \mid T) \approx \prod_{i=1}^{|S|} P(s_i \mid s_{i-1}, ..., s_{i-n+1}, \zeta_i) \quad (5)$$

$$P(S \mid T) \approx \prod_{i=1}^{|S|} P(s_i \mid s_{i+1}, ..., s_{i+n-1}, \zeta_i) \quad (6)$$

As the computational cost of NNLMs is a significant issue in decoding phase, we adopt two techniques for speeding up NNJM computation: self-normalization and pre-computation.

The self-normalization technique aims to avoid computing output softmax over the entire target vocabulary. Mainly, it replaces the training objective function with

$$L = \sum_i [\log(P(x_i)) - \alpha \log^2(Z(x_i))] \quad (7)$$

where $Z(x)$ is the summing part of softmax normalizer, and α is the parameter that controls trade-off between neural network accuracy and mean self-normalization error. At decoding phase, we simply use the input value of output layer as feature score, rather than $\log(P(x))$.

Another technique is called pre-computation, which computes dot product between the projection layer (word embedding) and the first hidden layer in advance. Furthermore, the computation of hyperbolic tangent (*tanh*) can also be accelerated using a lookup table.

In our experiments, we integrate the NNJM s2t.l2r model into our decoder, and the other variant models are used in the reranking step.

2.4. Domain Adaptation

Besides the TED portion data, the MultiUN [14] bilingual data can also be used for building translation models. However, the MultiUN Chinese-English parallel corpus provided by the IWSLT2014 Evaluation Campaign is aligned in chapter level. It cannot be used directly. To solve the problem, we firstly employ a tool hunalign [15] to automatically align the corpus at sentence-level.

In addition, the MultiUN data is almost 50 times larger than the in-domain parallel data, so it is unwise to treat them equally. We adopt a cross entropy based text selection method to choose partial volume from the MultiUN data [16]. In this method, an in-domain language model is applied to calculating cross entropy for each sentence pair, and then those with relatively low cross entropy are selected.

We select about 20% portion of the MultiUN data, and divide these data into several groups. For each group, we can train a translation model. There are two ways to incorporate these translation models into the system: linear interpolation and log-linear interpolation. We use the simple yet effective linear interpolation method. Each component probability in the translation model is linearly interpolated together. For example, let us consider the "backward" probability $p(s|t)$ of source language phrase *s* being generated by target language phrase *t*. For a set of $p_i(s|t)$, each trained on a sub-corpus, the mixture model is computed as

$$p(s|t) = \sum_{i=1}^{N} \alpha_i p_i(s|t) \quad (8)$$

To set the weights α_i, we firstly extract a set of phrase pairs from an in-domain development set using the training procedure. This yields a joint distribution $\tilde{p}$, which is used to define a maximum likelihood objective function as in Equation 9. The weights can then be learned efficiently using EM algorithm, which was first proposed in [17].

$$\tilde{\alpha} = \arg\max_{\alpha} \sum_{s,t} \tilde{p}_i(s,t) \log \sum_{i}^{N} \alpha_i p_i(s|t) \quad (9)$$

3. Experiments

In this section, we describe the experimental setup and results for both Chinese-English and English-Chinese translation tasks. We use the IWSLT 2013 test set for evaluating the techniques described above.

3.1. Chinese-English

As preprocessing, all the English texts in the corpora were tokenized by the tokenization tool in Moses toolkit. All Capital letters were converted to lower case. For Chinese, sentences need to be split into words. We compared several Chinese word segmentation tools and finally chose the in-house implementation. As post-processing, we use an SMT-based recaser to restore the true case for the output of the decoder. The experimental results are given in Table 1. All scores are case-sensitive BLEU.

3.1.1. Baseline Systems

Firstly, we built a weak baseline system ("*weak-baseline*" in Table 1) with the similar setup to that of the official baseline system in IWSLT 2013 [18]. All models are trained using the in-domain TED data provided by the campaign [19]. Bidirectional word alignments were trained by GIZA++ and symmetrized using *grow-diag-final-and* heuristic. An MSD-based lexical reordering model was applied. A 5-gram language model with modified Kneser-Ney smoothing was trained from the English part of the parallel corpus using SRILM toolkit. The weights of all features are optimized on dev2010 using MERT. Translation quality was evaluated on the tst2013 set in IWSLT 2013.

We obtained the strong baseline system by improving the following components: development set, word alignment, translation model, reordering model and weight tuning algorithm.

The official website released four sets for tuning, which are dev2010, tst2010, tst2011, and tst2012. Since bigger development set showed better performance in our pilot experiments, we combined them together and formed a big development set. Using the big development set for weight tuning gave rise to an improvement of +0.4% BLEU ("bigdev" in Table 1).

For word alignment, we improved GIZA++ with the l_0-norm. Although it has almost no effect on tst2013, it improved the development set by +0.16% BLEU. So we still keep it in our system. By simply replacing *grow-diag-final-and* by *grow*, our system gained further +0.14% BLEU.

There are only 180k sentence pairs in the TED training corpus, which is quite small. Over 90% phrase pairs in the phrase table occurred only once in the training corpus. This indicates data sparseness. Similarly to language model smoothing, we applied Good-Turing [20] to smoothing occurrence counts of phrase pairs, instead of using the counts directly. We obtained an improvement of +0.33% BLEU with Good-Turing smoothing ("GT smoothing" in Table 1).

As for the MSD based lexical reordering model, it is known that there are inconsistence about reordering orientation detection between training and decoding time [21]. A simple yet effective improvement is the hierarchical reordering model (HRM). Replacing MSD by HRM gave us another gain of +0.29% BLEU.

Finally, we adopted kbMIRA instead of MERT to tune feature weights. kbMIRA optimize BLEU less aggressively, improving model score and BLEU correlation across range of hypothesis. It produced an additional gain of +0.3% BLEU. Now we denote the system as "*strong-baseline*" in Table 1.

From "*weak-baseline*" to "*strong-baseline*", there are totally improvements of +1.45% BLEU on tst2013. Base on the "*strong-baseline*", we further improve our system by

adding more language models, neural network joint model, domain adapted translation models, etc.

3.1.2. Additional Features

Besides the parallel corpora, the official website also provides a number of monolingual English data. We used them to train n-gram language models. To be specific, each corpus was used to train a 5-gram language model with modified Kneser-Ney smoothing. Then we selected top ten language models according to the perplexity of LM on development set. Table 2 shows all of the selected corpora and the corresponding perplexities. The TED in-domain language model was the primary LM used in baseline systems and naturally has the lowest perplexity. We added these ten out-of-domain LMs to the decoder as separate features, and tuned their weights together with other features. We were surprising to see that these ten LMs gave us a great improvement up to +1.88% BLEU, which is the biggest improvement among all the techniques.

For NNJMs, we set up a projection layer of 192 dimensions and single hidden layer of 512 dimensions. Sizes of both input and output vocabularies are 10K. During training we set an initial learning rate of 10^{-3} and a mini-batch size of 128. Training was performed on GPU processor, and the decoding was carried out on CPU. By incorporating the s2t.l2r model into decoder, we achieved further gain of +0.5% BLEU.

MultiUN is the only out-of-domain parallel data that can be used in the campaign. It contains 9.5 million sentences, which is 52 times larger than the in-domain data. Instead of using all the MultiUN data, we selected about 1.9M parallel sentences from it using a cross-entropy based method [16], and divided them into four groups (125K, 250K, 500K, 1000K sentence pairs for each group). From each group, we trained one translation model. Then we linearly interpolated these models together with the in-domain model. Interpolation weights were trained by EM algorithm. This domain adaptation method improves performance by +0.18% BLEU (denoted by "*+UN_DA*").

In the last step, we tried to use more features to rerank k-best translations. We firstly generate 1000 best hypotheses from the "*+UN_DA*" system. Then five additional features were added for each hypothesis: three NNJM model (s2t.r2l, t2s.l2r, t2s.r2l) scores, a RNNLM score and a backward language model score. kbMIRA was used to tune weights for all features including those used in decoding. Reranking brought a further improve of +0.22% BLEU. The "reranking" result was our primary submission.

Table 1: Results for Chinese-English MT task

system	dev	tst2013
weak-baseline	10.61	14.19
+bigdev	13.20	14.59
+l_0-norm	13.36	14.58
+grow	13.42	14.72
+GT smoothing	13.65	15.05
+HRM	13.87	15.34
+ kbMIRA (strong-baseline)	13.91	15.64
+10 LMs	15.44	17.52
+NNJM	16.01	18.02
+UN_DA	16.20	18.20
+reranking	16.42	18.42

Table 2: Selected corpora for LMs and corresponding perplexities

data	bigdev
WIT[3] mono English (in-domain)	95.0
CzEng 1.0 from WMT14	103.7
News Crawl: 2013 from WMT14	104.8
News Crawl: 2012 from WMT14	107.4
News Crawl: 2011 from WMT14	108.9
nyt_eng from gigaword fifth edition	109.0
News Crawl: 2009 from WMT14	113.1
News Crawl: 2008 from WMT14	114.2
ltw_eng from gigaword fifth edition	116.8
News Crawl: 2010 from WMT14	117.4
News Crawl: 2007 from WMT14	128.6

Table 3: Results for English-Chinese MT task

System	bigdev	tst2013
	BLEU (char-based)	**BLEU** (char-based)
weak-baseline	14.92	18.87
strong-baseline	20.03	21.46
+wcLM	20.36	21.70
+OSM	20.47	22.05
+NNJM	20.83	22.35
+UN_DA	20.91	22.44
+reranking	21.01	22.55

3.2. English-Chinese

For the English-Chinese MT task, all the parallel and monolingual data are preprocessed exactly the same way as the Chinese-English task. All the scores showed in Table 3 are char-based BLEU. We also trained a weak baseline and a strong baseline using the same techniques as those in the Chinese-English task. The development set is also the same one, except that the source and target language are reversed. The "*strong-baseline*" achieves an improvement of +2.59% BLEU on tst2013 over the "*weak-baseline*".

Then, we improved the "*strong-baseline*" system by adding a 7-gram word class language model into the decoder (wcLM, +0.24% BLEU). All words were classified into 400 classes. After that, an Operation Sequence Model (OSM) was added. It gains +0.35% BLEU (Theses two techniques were also tried on the Chinese-English task, but no improvements were achieved. So we neglect them in the above sub-section). We also adopted NNJM (s2t.l2r, +0.31% BLEU) and domain adaptation for translation models (UN_DA, +0.09% BLEU). Finally, we reranked 1000-best hypotheses generated by "*+UN_DA*" system (reranking, +0.11% BLEU). The "reranking" result was our primary submission.

4. Conclusions

In this paper, we presented our submission runs and technical details of the IWSLT 2014 Evaluation Campaign in the optional MT track on Chinese-English and English-Chinese translations. The baseline system utilizes a state-of-the-art phrase-based translation decoder. After applying a lot of novel models and techniques, the translation results were significantly improved.

To summarize, main improvements result from the following techniques:

- Rich language model features. We build several large language models and integrate them into the log-linear model as separate features. We build different types of language models such as RNNLM, class-based LM and reverted-directional LM.
- Successfully implemented neural network models. We build NNJM, RNNLM for decoding or reranking, and achieve significant improvements.
- Effectively used data. We make a big development set by combining several previous test sets. Bigger development set produces better results. We extract some useful texts from MultiUN, which helps improve the translation model.

In the future, we are planning to integrate more features into our log-linear models.

5. References

[1] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL), Sapporo, Japan, July 2003, pp. 160–167.

[2] A. Vaswani, L. Huang, and D. Chiang, Smaller alignment models for better translations: unsupervised word alignment with the l_0-norm. In Proc. ACL, 311–319, 2012.

[3] C. Cherry and G. Foster. 2012. Batch tuning strategies for statistical machine translation. In HLT-NAACL, pages 427–436, Montr´ eal, Canada, June.

[4] Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A Joint Sequence Translation Model with In tegrated Reordering. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 1045–1054, Portland, Oregon, USA, June.

[5] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL) on Interactive Poster and Demonstration Sessions, pages 177-180.

[6] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, ser. EMNLP '08. Stroudsburg, PA, USA:Association for Computational Linguistics, 2008, pp. 848–856.

[7] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in Proc. of the Int. Conf. on Speech and Language Processing (ICSLP), vol. 2, Denver, CO, Sept. 2002, pp. 901–904.

[8] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA,July 2002, pp. 311–318.

[9] Deyi Xiong, Min Zhang, Haizhou Li: Enhancing language models in statistical machine translation with backward n-grams and mutual information triggers. ACL-HLT 2011: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, June 19-24, 2011; pp.1288-1297.

[10] J. Wuebker, S. Peitz, F. Rietig, and H. Ney, "Improving statistical machine translation with word class models," in Conference on Empirical Methods in Natural Language Processing, Seattle, USA, Oct. 2013, pp. 1377–1381.

[11] Tomas Mikolov, Stefan Kombrink, Lukas Burget, JH Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 5528-5531.

[12] Holger Schwenk. 2010. Continuous-space Language Models for Statistical Machine Translation. Prague Bull. Math. Linguistics, 93:137-146.

[13] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In 52nd Annual Meeting of the Association for Computational Linguistics, pages 1370-1380.

[14] A. Eisele and Y. Chen, "MultiUN: A Multilingual Corpus from United Nation Documents," in Proceedings of the Seventh conference on International Language Resources and Evaluation, May 2010, pp. 2868–2872.

[15] D. Varga, L. Nemeth, P. Halacsy, A. Kornai, Viktor Tron, and V. Nagy. 2005. Parallel corpora for medium density languages. In RANLP, pages 560–596, Borovets, Bulgaria.

[16] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK., July 2011, pp. 355–362.

[17] George Foster, Cyril Goutte and Roland Kuhn. 2010. Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. EMNLP. Cambridge, MA.

[18] M. Cettolo, J. Niehues, S. Stker, L. Bentivogli, and M. Federico, "Report on the 10th iwslt evaluation campaign," in Proceedings of the 10th International Workshop on Speech Language Translation, 2013.

[19] M. Cettolo, C. Girardi, and M. Federico. 2012. WIT3: Web Inventory of Transcribed and Translated Talks. In Proc. of EAMT, pp. 261-268, Trento, Italy.

[20] G. Foster, R. Kuhn, and H. Johnson. "Phrasetable smoothing for statistical machine translation". Proc. EMNLP, pp. 53-61, Sydney, Australia, July 2006

[21] C. Tillmann. 2004. "A Unigram Orientation Model for Statistical Machine Translation". NAACL.

The NICT Translation System for IWSLT 2014

Xiaolin Wang *Andrew Finch* *Masao Utiyama* *Taro Watanabe* *Eiichiro Sumita*

Multilingual Translation Group
National Institute of Information and
Communications Technology
Kyoto, Japan
{first.last}@nict.go.jp, mutiyama@nict.go.jp

Abstract

This paper describes NICT's participation in the IWSLT 2014 evaluation campaign for the TED Chinese-English translation shared-task. Our approach used a combination of phrase-based and hierarchical statistical machine translation (SMT) systems. Our focus was in several areas, specifically system combination, word alignment, and various language modeling techniques including the use of neural network joint models. Our experiments on the test set from the 2013 shared task, showed that an improvement in BLEU score can be gained in translation performance through all of these techniques, with the largest improvements coming from using large data sizes to train the language model.

1. Introduction

In the IWSLT 2014 machine translation evaluation campaign, the NICT team participated in the TED [1] translation shared-task for Chinese-English. This paper describes the machine translation approach adopted for this campaign.

Our system was a combination of phrase-based and hierarchical SMT systems. The combination was performed by reranking the n-best hypotheses from these systems. A log-linear model which used the hypothesis scores of the component systems as features was used to calculate the score used in reranking. Additional features were also added into the log-linear model, for example features from a neural network model, or talk-level language model scores.

In addition to system combination, we put emphasis on language modeling. We used three approaches to improve the language modeling in the system. In the first approach we used a language model that was an interpolation of an in-domain language model, and a language model trained on the GIGAWORD data. In the second approach, we incorporated a language model trained on the machine translations of each talk in the test dataset into the reranking procedure. In the third approach, a bilingual feed-forward neural network [2] was used in the reranker.

Finally, we also improved the word alignment by using combining the alignments from two independent aligners: GIZA++ [3] and a modified version of the CICADA aligner [4].

2. Data

We used same Chinese-English data sets in all of the experiments in this paper. The supplied bilingual data consisted of 179901 sentence pairs. From this data we randomly selected a 3023-pair development set for tuning the decoder, and a 1553-pair development set for tuning the reranker. These development sets consisted of complete talks. All of the remaining talks were used as bilingual training data for the component SMT systems. We used the IWSLT 2013 test set for evaluation. For some of the experiments we used language models trained on the English LDC Gigaword dataset, a collection of approximately 4 billion words of international newswire text.

2.1. Pre-processing

The English data was tokenized by applying the EUROPARL tokenizer [5]. We also removed all case information from the English text to help to minimize issues of data sparseness in the models of the translation system. All punctuation was left in both source and target. We took the decision to generate target punctuation directly using the process of translation, rather than as a punctuation restoration step in post processing based on experiments carried out for the 2010 IWSLT shared evaluation [6].

2.2. Post-processing

The output of the translation system was subject to the following post-processing steps which were carried out in the following order:

1. In all experiments, the out of vocabulary words (OOVs) were passed through the translation process unchanged, some of these OOVs were Chinese and some English. For the primary submission, we took

the decision to delete only those OOVs containing Chinese characters not included in the ASCII character set and leave words containing only ASCII characters in the output.

2. The output was de-tokenized using the de-tokenizer included with the MOSES toolkit [7].

3. The output was re-cased using the re-casing tool supplied with the MOSES toolkit. We trained the re-casing tool on cased text from the TED talk training data.

3. The Base Systems

3.1. Decoders

Our submission used two SMT systems within a system combination framework; these systems were:

1. OCTAVIAN, an in-house phrase-based decoder.

2. A hierarchical version of the MOSES decoder [7].

The OCTAVIAN decoder used in these experiments is an in-house phrase-based statistical machine translation decoder that can operate in a similar manner to the publicly available MOSES decoder [7]. The base decoder used a standard set of features that were integrated into a log-linear model using independent exponential weights for each feature. These features consisted of: a language mode; five translation model features; a word penalty; and a lexicalized re-ordering model with monotone, discontinuous, swap features for the current and previous phrase-pairs. We decoded with a reordering limit of 5 in the OCTAVIAN phrase-based decoder.

3.2. Language Model Training

The language models were built using the SRI Language Modeling Toolkit [8]. A 5-gram model was built for decoding the development and test data for evaluation. The language models were smoothed using modified Knesser-Ney smoothing.

3.3. Translation Model Training

The translation model for the base system was built in the standard manner using a 2-step process. First the training data was word-aligned using a combination of the CICADA and GIZA++ [3] aligners. Two copies of the corpus were aligned independently with each aligner, then the aligned copies were concatenated prior to phrase extraction. Second, the phrase-extraction heuristics from the MOSES [7, 9] machine translation toolkit were used to extract a set of bilingual phrase-pairs using the alignments.

3.4. Parameter Tuning

To tune the values for the log-linear weights in our system, we used the standard minimum error-rate training procedure (MERT) [10]. The weights for the models were tuned using the development data supplied for the task.

Component System	BLEU (%)
OCTAVIAN	14.74
MOSES (hierarchical)	14.95

Table 1: BLEU scores of the component systems

3.5. Evaluation

We evaluated each of these systems on the IWSLT 2013 test set, and the results are shown in Table 3.5. The evaluation in all of the experiments in this report was carried out on tokenized, lowercase data, using the "multi-bleu.perl" evaluation script included in release version 2.1 of the MOSES toolkit. The systems are roughly comparable in performance, and about 1.5 BLEU percentage points higher than the case-insensitive MOSES baseline reported in [11], we believe this can be explained by differences in the tokenization used for evaluation, and also by differences in the development sets used for tuning. We found that when tuned and evaluated on different data sets, the relative rankings of the systems may vary.

4. Methodology

4.1. Language Modeling

4.1.1. Neural Network Model

We implemented the neural network joint models proposed in [2] and used the output as a feature in the reranker. We ran a set of experiments to determine the optimal network architecture. We varied the size of the context on both source and sides, and also the scale of the neural network. We found the settings used in [2] gave rise the highest performance, and we therefore adopted these settings in our system. These settings were: 11-word source context, 3-word target context, 192-unit shared embedding layer, and two additional 512-unit hidden layers. We set both input and output vocabulary size to 32000. The neural network was implemented using the NPLM toolkit [12].

The results are shown in Table 4.3. The gain using from this approach was approximately 0.5 BLEU points. This was lower than the gains reported in [2], however, in their experiments the neural network was directly integrated into the decoding process. We integrated monolingual neural network model into the OCTAVIAN decoder, however, the experiments were not completed due to time limitations.

4.1.2. Gigaword

We combined language models trained on the source of the parallel TED corpus, and the Gigaword newswire corpus by linear interpolation. The interpolated language model was then used directly in the decoding process, and constituted a

SMT System	BLEU (%)
OCTAVIAN TED LM	14.74
OCTAVIAN TED+Gigaword	16.72
MOSES hierarchical TED LM	14.95
MOSES hierarchical TED+Gigaword	16.83

Table 2: Evaluation of the effectiveness of using a large out-of-domain language model.

single feature in the log-linear model. The interpolation was done using the SRI Language Modeling Toolkit [8]. We ran pilot experiments to determine the best interpolation weight by grid search and found a weight of 0.5 to be the most effective. Both of the language models were trained with modified Knesser-Ney smoothing [13, 14].

The results are shown in Table 4.1.2. It is clear that adding a large out-of-domain language model is very effective on our task.

4.1.3. Talk-level Model

This model was a language model built by applying the SRI Language Modeling Toolkit to machine translated output. The talk-level language model was built from the set of 1000-best translation hypotheses obtained by translating the test set using each of the component translation systems. The 1000-best lists from the component systems were merged, into a set of unique word sequences. A different language model was build from each talk in the test set, and applied only to sentences from the same talk. The score of the language model was used as a feature for reranking.

The results are shown in Table 4.1.2 and show a modest improvement in performance over the baseline without this model.

4.2. Alignment

Two copies of the training data were aligned. One copy with GIZA++, and the other with an enhanced version of the CICADA aligner. The SMT models derived from the alignment were trained on the union of this aligned data.

The results are shown in Table 4.2. The largest gain arises from using the CICADA aligner together with the hierarchical SMT system. However we took the decision to use this strategy in our primary submission because in pilot experiments the strategy based on a combination of methods typically outperformed the strategy based on a single method.

4.3. System Combination

The system combination was performed by integrating features from the component SMT systems, together with a set of additional features within the framework of a log-linear model. The log-linear weights of all the features were tuned on a separate development set using the same MERT approach as in tuning the weights in the models used by the decoders. The features using in reranking were:

SMT System	BLEU (%)
OCTAVIAN GIZA++	14.74
OCTAVIAN CICADA	15.21
OCTAVIAN Union	15.22
MOSES hierarchical GIZA++	14.95
MOSES hierarchical CICADA	15.56
MOSES hierarchical Union	15.54

Table 3: Evaluation of the various alignment strategies.

SMT System	BLEU (%)
OCTAVIAN baseline	17.09
MOSES hierarchical baseline	17.56
Combination	17.65
Combination with neural network joint model	17.88
Combination with talk-level LM	17.68
Combination with all features	17.92

Table 4: Evaluation of the combination systems.

1. The decoder score from the OCTAVIAN decoder;
2. The decoder score from the hierarchical MOSES decoder;
3. The output from the joint neural language model;
4. The talk-level language model score.

1000-best lists from the 2-component systems were merged in the following manner:

1. The n-best lists of each component system were made unique; only the best scoring hypotheses was kept from a set of duplicate hypotheses which gave rise to the same target word sequence.
2. Hypotheses with the target text were merged across systems into a single hypothesis, receiving the respective decoder scores in features 1. and 2.
3. If the hypothesis was only generated by one of the component systems, it received zero for the feature corresponding to the decoder that did not generate it.
4. Features 3. and 4. were then calculated for each hypothesis.

The results are shown in Table 4.3. Both of the component systems used in the combination were trained using the enhanced alignment method proposed in Section 4.2, and included the interpolated language model described in Section 4.1.2.

5. Conclusions

This paper described NICT's system for the IWSLT 2014 evaluation campaign for the TED Chinese-English translation shared-task. Our approach was based on a combination of hierarchical and phrase-based statistical machine translation systems integrated with other features within the framework of a single log-linear model. We augmented the base systems using multiple alignment strategies, a neural network joint model, and a talk-level language model. We were able to improve the translation performance over a phrase-based MOSES baseline without these features by 2.96 BLEU points.

6. References

[1] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[2] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. M. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, 2014, pp. 1370–1380. [Online]. Available: http://aclweb.org/anthology/P/P14/P14-1129.pdf

[3] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[4] T. Watanabe, "The cicada open source aligner, http://www2.nict.go.jp/univ-com/multi_trans/cicada/."

[5] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT summit*, vol. 5, 2005, pp. 79–86.

[6] C.-L. Goh, T. Watanabe, M. Paul, A. Finch, and E. Sumita, "The NICT Translation System for IWSLT 2010," in *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, M. Federico, I. Lane, M. Paul, and F. Yvon, Eds., 2010, pp. 139–146.

[7] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowa, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007): demo and poster sessions*, Prague, Czeck Republic, June 2007, pp. 177–180.

[8] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," in *Proceedings of the International Conference on Spoken Language Processing*, vol. 2, Denver, 2002, pp. 901–904.

[9] P. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," in *Machine translation: from real users to research: 6th conference of AMTA*, Washington, DC, 2004, pp. 115–124.

[10] F. J. Och, "Minimum error rate training for statistical machine translation," in *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL 2003)*, Sapporo, Japan, 2003.

[11] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 10th iwslt evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Heidelberg, Germany, December 2013, pp. 29–38.

[12] A. Vaswani, Y. Zhao, V. Fossum, and D. Chiang, "Decoding with large-scale neural language models improves translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2013, pp. 1387–1392. [Online]. Available: http://aclweb.org/anthology/D/D13/D13-1140.pdf

[13] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1. IEEE, 1995, pp. 181–184.

[14] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1996, pp. 310–318.

Polish - English Speech Statistical Machine Translation Systems for the IWSLT 2014.

Krzysztof Wołk, Krzysztof Marasek

Multimedia Department
Polish Japanese Academy of Information Technology, Koszykowa 86, 02-008 Warsaw
kwolk@pja.edu.pl, kmarasek@pja.edu.pl

Abstract

This research explores effects of various training settings between Polish and English Statistical Machine Translation systems for spoken language. Various elements of the TED parallel text corpora for the IWSLT 2014 evaluation campaign were used as the basis for training of language models, and for development, tuning and testing of the translation system as well as Wikipedia based comparable corpora prepared by us. The BLEU, NIST, METEOR and TER metrics were used to evaluate the effects of data preparations on translation results. Our experiments included systems, which use lemma and morphological information on Polish words. We also conducted a deep analysis of provided Polish data as preparatory work for the automatic data correction and cleaning phase.

1. Introduction

Polish is one of the complex West-Slavic languages, which represents a serious challenge to any SMT system. The grammar of the Polish language, with its complicated rules and elements, together with a big vocabulary (due to complex declension) are the main reasons for its complexity (in Polish there are seven cases, three genders, animate and inanimate nouns, adjectives agreed with nouns in terms of gender, case and number and a lot of words borrowed from other languages which are often inflected similarly to those of Polish origin).

This greatly affects the data and data structure required for statistical models of translation. The lack of available and appropriate resources required for data input to SMT systems presents another problem. SMT systems should work best in specified, not too wide text domains and will not perform well for general use. Good quality parallel data, especially in a required domain has low availability. In general, Polish and English differ also in syntax. English is a positional language, which means that the syntactic order (the order of words in a sentence) plays a very important role, particularly due to limited inflection of words (e.g. lack of declension endings). Sometimes, the position of a word in a sentence is the only indicator of the sentence meaning. In the English sentence, the subject group comes before the predicate, so the sentence is ordered according to the Subject-Verb-Object (SVO) schema. In Polish, however, there is no specific word order imposed and the word order has no decisive influence on the understanding of the sentence. One can express the same thought in several ways, which is not possible in English. For example, the sentence „I just tasted a new orange juice." can be written in Polish as „Spróbowałem właśnie nowego soku pomarańczowego", or "Nowego soku pomarańczowego właśnie spróbowałem.", or "Właśnie spróbowałem nowego soku pomarańczowego.", or „Właśnie nowego soku pomarańczowego spróbowałem." Differences in potential sentence orders make the translation process more complex, especially when working on a phrase-model with no additional lexical information.

As a result starting point was much lower than for other languages, however our progress in last 3 years was faster than others [1,2]. The aim of this work is to create an SMT system for translation from Polish to English (and the other way round, i.e. from English to Polish) to address the IWSLT 2014 [3] evaluation campaign requirements. This paper is structured as follows: Section 2 explains the Polish data preparation. Section 3 presents the English language issues. Section 4 describes the translation evaluation methods. Section 5 presents the results. Lastly in Section 6 we summarize potential implications and ideas for future work.

2. Preparation of the Polish data

The Polish data in the TED talks (about 17 MB) include almost 2,5 million words that are not tokenized. The transcripts themselves are provided as pure text encoded with UTF-8 and the transcripts are prepared by the IWSLT team [4]. In addition, they are separated into sentences (one per line) and aligned in language pairs.

It should be emphasized that both automatic and manual preprocessing of this training information was required. The extraction of the transcription data from the provided XML files ensured an equal number of lines for English and Polish. However, some of the discrepancies in the text parallelism could not be avoided. These discrepancies are mainly repetitions of the Polish text not included in the English text.

Another problem was that TED 2013 data was full of errors. [5]. For the IWSLT 2014 we helped in repairing those errors in train, test and development sets. It was done semi-automatically by the usage of our tool described in [6]. We repaired spelling errors that artificially increased the dictionary size in Polish side of the corpora. Additionally we filtered out and repaired bi-sentences with odd nesting, such as:

Part A, Part A, Part B, Part B.

e.g.

"Ale będę starał się udowodnić, że mimo złożoności, Ale będę starał się udowodnić, że mimo złożoności, istnieją pewne rzeczy pomagające w zrozumieniu. istnieją pewne rzeczy pomagające w zrozumieniu."

Some parts (words or full phrases or even whole sentences) were duplicated. Furthermore, there were segments containing repetitions of whole sentences inside a single segment. For instance:

Sentence A. Sentence A.

e.g.

"Zakumulują się u tych najbardziej pijanych i skąpych. Zakumulują się u tych najbardziej pijanych i skąpych."

or

Part A, Part B, Part B, Part C

e.g.

" Matka może się ponownie rozmnażać, ale jak wysoką cenę płaci, przez akumulację toksyn w swoim organizmie - przez akumulację toksyn w swoim organizmie - śmierć pierwszego młodego."

Overall, in the train set we found about 7% of spelling errors and about 15% of insertion errors. Luckily such problems occur only on the Polish side of the corpora. In our opinion the pre-processing tools used to align the corpus were not adjusted for the Polish language. Cleaning those problems increases BLEU score by the factor of 1,5 – 2.

The number of unique Polish words and their forms was 144,115 and 59,296 English unique word forms. The disproportionate vocabulary sizes are also a challenge especially in translation from English to Polish.

Another problem is that the TED Talks do not have any specific domain. Statistical Machine Translation by definition works best when very specific domain data is used. The data we have is a mix of various, unrelated topics. This is most likely the reason why we cannot expect big improvements with this data and generally low scores in translation quality metrics.

There is not much focus on Polish in the campaign, so there is almost no additional data in Polish in comparison to a huge amount of data in, for example, French or German. At first we used perplexity measurement metrics to determine the data we obtained. Some of the data we were able to obtain from the OPUS [12] project page, some from another small projects and the rest was collected manually using web crawlers. We created those corpora and used them. What we created was:

- A Polish – English dictionary (bilingual parallel)
- Additional (newer) TED Talks data sets not included in the original train data (we crawled bilingual data and created a corpora from it) (bilingual parallel)
- E-books (monolingual PL + monolingual EN)
- Proceedings of UK House of Lords (monolingual EN)
- Subtitles for movies and TV series (monolingual PL)
- Parliament and senate proceedings (monolingual PL)
- Wikipedia Comparable Corpus (bilingual parallel)
- Euronews Comparable Corpus (bilingual parallel)
- Repository of PJIIT's diplomas (monolingual PL)
- Many PL monolingual data web crawled from main web portals like blogs, chip.pl, Focus newspaper archive, interia.pl, wp.pl, onet.pl, money.pl, Usenet, Termedia, Wordpress web pages, Wprost newspaper archive, Wyborcza newspaper archive, Newsweek newspaper archive, etc.

"Other" in the table below stands for many very small models merged together. In Table 1 we show the perplexity values of the obtained data with no smoothing (PPL in Table 1) as well as smoothed with the Kneser-Ney algorithm (PPL+KN in Table 1). We used the MITLM [29] toolkit for that evaluation. As an evaluation set we used dev2010 data, which was used for tuning. Its dictionary covers 2861 words.

EMEA are texts from the European Medicines Agency, KDE4 is a localization file of that GUI, ECB stands for European Central Bank corpus, OpenSubtitles [12] are movies and TV series subtitles, EUNEWS is a web crawl of the euronews.com web page and EUBOOKSHOP comes from bookshop.europa.eu. Lastly bilingual TEDDL is additional TED data. We ensured that this data was not overlapping with the test or development sets. As can be seen from the Table 1, all additional data has big perplexity values, so no astonishing improvements based only on data could be expected.

Table 1: Data Perplexities for dev2010 data set

Data set	Dictionary	PPL	PPL + KN
Baseline train.en	44,052	221	223
EMEA	30,204	1738	1848
KDE4	34,442	890	919
ECB	17,121	837	889
OpenSubtitles	343,468	388	415
EBOOKS	528,712	405	417
EUNEWS	21,813	430	435
NEWS COMM	62,937	418	465
EUBOOKSHOP	167,811	921	950
UN TEXTS	175,007	681	714
UK LORDS	215,106	621	644
NEWS 2010	279,039	356	377
GIGAWORD	287,096	582	610
DICTIONARY	39,214	8629	8824
OTHER	13,576	492	499
WIKIPEDIA	682,276	9131	9205
NEWSPAPERS	608,186	10066	10083
WEB PORTALS	510,240	731	746
BLOGS	76,697	3481	3524
USENET	733,619	8019	8034
DIPLOMAS	353,730	32345	32582
TEDDL	47,015	277	277

WIKIPEDIA and EUNEWS are parallel corpora extracted by us from comparable corpora. We were able to obtain 4,498 topic-aligned articles from the Euronews and about 1M from the Wikipedia. The Wikipedia corpus was about 104MB in size and contained 475,470 parallel sentences. Its first version was acknowledged as permissible data for the IWSLT 2014 evaluation campaign. The Euronews corpora contained 1,617 bi-sentences.

In order to extract the parallel sentence pairs we decided to facilitate Yalign Tool [26]. The Yalign tool was designed in order to automate parallel text mining process by finding sentences that are close translation matches from the comparable corpora. This opened up avenues for harvesting parallel corpora from sources like translated documents and the web. What is more Yalign is not limited to any language pair. But creation of own alignment models for two required languages is necessary.

The Yalign tool was implemented using a sentence similarity metric that produces a rough estimate (a number between 0 and 1) of how likely it is for two sentences to be a translation of each other. Additionally it uses a sequence aligner, that produces an alignment that maximizes the sum of the individual (per sentence pair) similarities between two documents. Yalign's main algorithm is actually a wrapper before standard sequence alignment algorithm [26].

For the sequence alignment Yalign uses a variation of the Needleman-Wunch algorithm [27] to find an optimal alignment between the sentences in two given documents. The algorithm has polynomial time worst-case complexity and it produces an optimal alignment. Unfortunately it can't handle alignments that cross each other or alignments from two sentences into a single one [27].

Since the sentence similarity is a computationally expensive operation, the implemented variation of the Needleman-Wunch algorithm uses A* approach to explore the search space instead of using the classical dynamic programming method that would require N * M calls to the sentence similarity matrix.

After the alignment, only sentences that have a high probability of being translations are included in the final alignment. The result is filtered in order to deliver high quality alignments. To do this, a threshold value is used, such that if the sentence similarity metric is low enough the pair is excluded.

For the sentence similarity metric the algorithm uses a statistical classifier's likelihood output and adapts it into the <0,1> range.

The classifier must be trained in order to determine if a pair of sentences is translation of each other or not. The particular classifier used in the Yalign project was a Support Vector Machine. Besides being excellent classifier, SVMs can provide a distance to the separation hyperplane during classification, and this distance can be easily modified using a Sigmoid Function to return likelihood between 0 and 1 [28].

The use of a classifier means that the quality of the alignment depends not only on the input but also on the quality of the trained classifier.

To train the classifier a good quality parallel data was necessary as well as a dictionary with translation probability included. For this purposes we used TED talks [3] corpora enhanced by us during the IWSLT'13 Evaluation Campaign [5]. In order to obtain a dictionary we trained a phrase table and extracted 1-grams from it. We used the MGIZA++ tool for word and phrase alignment. The lexical reordering was set to use the msd-bidirectional-fe method and the symmetrisation method was set to grow-diag-final-and for word alignment processing [5].

Before use of a training translation model, preprocessing that included removal of long sentences (set to 80 words) had to be performed. The Moses toolkit scripts [7] were used for this purpose.

The final processing corpus included 185,527 lines from the Polish to English corpus. However, the disproportionate vocabulary sizes remained. One of the solutions to this problem (according to work of Bojar [10]) was to use stems instead of surface forms in order to reduce the Polish vocabulary size. Such a solution also requires a creation of an SMT system from Polish stems to plain Polish. Subsequently, we used PSI-TOOLKIT [9] to convert each Polish word into a lemma. The toolkit is a tool chain for automatic processing of Polish language and to lesser extent other languages like English, German, French, Spanish and Russian (with the focus on machine translation). The tool chain includes segmentation, tokenization, lemmatization, shallow parsing, deep parsing, rule-based machine translation, statistical machine translation, automatic generation of inflected forms from lemma sequences and automatic post edition. The toolkit was used as an additional information source for the SMT system preparation. It can be also used as a first step for implementing a factored SMT system that, unlike a phrase-based system, includes morphological analysis, translation of lemmas and features as well as generation of surface forms. Incorporating additional linguistic information should effectively improve translation performance [8].

2.1. Polish lemma extraction

As previously mentioned, lemma extracted from Polish words are used instead of surface forms to overcome the problem of the huge difference in vocabulary sizes. For Polish lemma extraction, a tool chain that included tokenization and lemmatization from PSI-TOOLS was used.

These tools used in sequence provide a rich output that includes a lemma form of the tokens, prefixes, suffixes and morphosyntatic tags. Unfortunately unknown words like names or abbreviations or numbers, etc. are lost in the process. Also capitalization as well as punctuation does not remain. To preserve this relevant information we implemented a specialized tool that basing on differences between input and output of the PSI-TOOLS restored most of the lost information. The lemmatized version of the Polish training data was reduced to 36,065 unique words and the polish language model was also reduced from 156,970 to 32,873 unique words. The results of this work are presented in Table 2 and in Table 3. Each experiment was done only on the baseline data sets in PL->EN and EN->PL direction. The system settings are described in Chapter 5. The year column shows the test set that was used in the experiment, if a year has L suffix in means that it is lemmatized version of the baseline system.

Table 2: PL Lemma to EN translation results

YEAR	BLEU	NIST	TER	MET
2010	16,70	5,70	67,83	49,31
2010L	13,33	4,68	70,86	46,18
2011	20,40	5,71	62,99	53,13
2011L	16,21	5,11	67,16	49,64
2012	17,22	5,37	65,96	49,72
2012L	13,29	4,64	69,59	45,78
2013	18,16	5,44	65,50	50,73
2013L	14,81	4,88	68,96	47,98
2014	14,71	4,93	68,20	47,20
2014L	11,63	4,37	71,35	44,55

Table 3: EN to PL Lemma translation results

YEAR	BLEU	NIST	TER	MET
2010	9,95	3,89	74,66	32,62
2010L	12,98	4,86	68,06	40,19
2011	12,56	4,37	70,13	36,23
2011L	16,36	5,40	62,96	44,86
2012	10,77	3,92	75,79	33,80
2012L	14,13	4,83	69,76	41,52
2013	10,96	3,91	75,95	33,85
2013L	15,21	5,02	68,17	42,58
2014	9,29	3,47	82,58	31,15
2014L	12,35	4,44	75,27	39,12

Our experiments show that lemma translation to EN in each test set decreased the evaluation scores, contrary translation from EN to lemma for each set increased the translation quality. Such solution requires also training of a system from lemma into PL in order to restore proper surface

forms of the words. We trained such system as well and evaluated it on official tests sets from years 2010-2014 and tuned on 2010 development data. The results for that system are presented in Table 4. Even that the scores are relatively high the results do not seem to be satisfactory enough to provide overall improvement of EN-LEMMA-PL pipeline over direct translation from EN to PL.

Table 4: Lemma to PL translation results

YEAR	BLEU	NIST	TER	MET
2010	41,14	8,72	31,28	65,25
2011	41,68	8,68	30,64	65,99
2012	38,87	8,38	32,23	64,18
2013	40,27	8,30	31,67	64,44
2014	37,78	8,01	33,17	62,78

To confirm our prediction we conducted additional experiment in which the English sentences were first translated into lemma and secondly we translated lemma into Polish surface forms. The results of such combined translation are showed in Table 5. They decrease the translation quality in comparison to direct translation from EN to PL. What is more by lemmatizing PL we lost much significant information. As a part of the future work we intend to lemmatize only not very common words, but we are still aware of that most of the Polish words will appear quire rare due to many word forms. We anticipate that most of the words will be replaced by lemmas. Unfortunately also the quality of lemma to surface is of low quality. The Polish declension is complex e.g. sometimes even a steam is changed doe to phonetic/phontactic rules.

Table 5: EN -> PL Lemma -> PL pipeline translation

YEAR	BLEU	NIST	TER	MET
2010	7,47	3,45	76,17	29,16
2011	9,67	3,84	72,45	32,25
2012	8,26	3,39	78,40	29,60
2013	8,83	3,54	77,11	30,61
2014	6,98	3,10	83,81	27,71

3. English Data Preparation

The preparation of the English data was definitively less complicated than for Polish. We developed a tool to clean the English data by removing foreign words, strange symbols, etc. Compare to Polish, the English data contained significantly less errors. Nonetheless some problems needed to be removed, most problematic were translations into languages other than English and strange UTF-8 symbols. We also found few duplications and insertions inside single segments.

4. Evaluation Methods

Metrics are necessary to measure the quality of translations produced by the SMT systems. For this, various automated metrics are available to compare SMT translations to high quality human translations. Since each human translator produces a translation with different word choices and orders, the best metrics measure SMT output against multiple reference human translations. For scoring purposes we used four well-known metrics that show high correlation with human judgments. Among the commonly used SMT metrics are: Bilingual Evaluation Understudy (BLEU), the U.S. National Institute of Standards & Technology (NIST) metric, the Metric for Evaluation of Translation with Explicit Ordering (METEOR), and Translation Error Rate (TER).

According to Koehn, BLEU [11] uses textual phrases of varying length to match SMT and reference translations. Scoring of this metric is determined by the weighted averages of those matches. [13]

To encourage infrequently used word translation, the NIST [13] metric scores the translation of such words higher and uses the arithmetic mean of the n-gram matches. Smaller differences in phrase length incur a smaller brevity penalty. This metric has shown advantages over the BLEU metric.

The METEOR [13] metric also changes the brevity penalty used by BLEU, uses the arithmetic mean like NIST, and considers matches in word order through examination of higher order n-grams. These changes increase score based on recall. It also considers best matches against multiple reference translations when evaluating the SMT output.

TER [14] compares the SMT and reference translations to determine the minimum number of edits a human would need to make for the translations to be equivalent in both fluency and semantics. The closest match to a reference translation is used in this metric. There are several types of edits considered: word deletion, word insertion, word order, word substitution, and phrase order.

5. Experimental Results

A number of experiments were performed to evaluate various versions for our SMT systems. The experiments involved a number of steps. Processing of the corpora was accomplished, including tokenization, cleaning, factorization, conversion to lower case, splitting, and a final cleaning after splitting. Training data was processed, and the language model was developed. Tuning was performed for each experiment. Lastly, the experiments were conducted.

The baseline system testing was done using the Moses open source SMT toolkit with its Experiment Management System (EMS) [15]. The SRI Language Modeling Toolkit (SRILM) [19] with an interpolated version of the Kneser-Key discounting (interpolate –unk –kndiscount) was used for 5-gram language model training. We used the MGIZA++ tool for word and phrase alignment. KenLM [17] was used to binarize the language model, with a lexical reordering set to use the msd-bidirectional-fe model. Reordering probabilities of phrases are conditioned on lexical values of a phrase. It considers three different orientation types on source and target phrases like monotone(M), swap(S) and discontinuous(D). The bidirectional reordering model adds probabilities of possible mutual positions of source counterparts to current and following phrases [18]. MGIZA++ is a multi-threaded version of the well-known GIZA++ tool [20]. The symmetrization method was set to grow-diag-final-and for word alignment processing. First two-way direction alignments obtained from GIZA++ were intersected, so only the alignment points that occurred in both alignments remained. In the second phase, additional alignment points existing in their union were added. The growing step adds potential alignment points of unaligned words and neighbors. Neighborhood can be set directly to left, right, top or bottom, as well as to diagonal (grow-diag). In the final step, alignment points between words from which at least one is unaligned are

added (grow-diag-final). If the grow-diag-final-and method is used, an alignment point between two unaligned words appears. [15]

We conducted about a hundred of experiments using test and development 2010 data to determine the best possible translation settings from Polish to English and the reverse. For experiments we used Moses SMT with Experiment Management System (EMS) [15]. Starting from baseline (BLEU: 16,70) system tests, we raised our score through extending the language model with more data and by interpolating it linearly. We determined that not using lower casing, changing maximum sentence length to 95, maximum phrase length to 6 improves the BLEU score. Additionally we changed the language model order from 5 to 6 and changed the discounting method from Kneser-Ney to Witten-Bell. Those setting proved to increase translation quality for PL-EN language pair in [5]. In the training part, we changed the lexicalized reordering method from msd-bidirectional-fe to hier-mslr-bidirectional-fe. The system was also enriched with Operation Sequence Model (OSM) [21]. The motivation for OSM is that it provides phrase-based SMT models the ability to memorize dependencies and lexical triggers, it can search for any possible reordering, and it has a robust search mechanism. Additionally, OSM takes source and target context into account, and it does not have the spurious phrasal segmentation problem. The OSM is valuable especially for the strong reordering mechanism. It couples translation and reordering, handles both short and long distance reordering, and does not require a hard reordering limit [21]. What is more we used Compound Splitting feature [8]. Tuning was done using MERT tool with batch-mira feature and n-best list size was changed from 100 to 150. This setting and language models produced the score of BLEU equal to 21,57. Lastly we used all parallel data we were able to obtain. We adapted it using Modified Moore Levis Filtering [8]. From our experiments we conducted that best results are obtained when sampling about 150,000 bi-sentences from in-domain corpora and by using filtering after the word alignment. The ratio of data to be kept was set to 0,8 obtaining our best score equal to 23,74.

Because of a much bigger dictionary, the translation from EN to PL is significantly more complicated. Our baseline system score was 9,95 in BLEU. Similarly to PL-EN direction we determined that not using lower casing, changing maximum sentence length to 85, maximum phrase length to 7 improves the BLEU score. Additionally we set the language model order from 5 to 6 and changed the discounting method from Kneser-Ney to Witten-Bell. In the training part, we changed the lexicalized reordering method from msd-bidirectional-fe to tgttosrc. The system was also enriched with Operation Sequence Model (OSM). What is more we used Compund Splitting feature and we did punctuation normalization. Tuning was done using MERT tool with batch-mira feature and n-best list size was changed from 100 to 150. Training a hierarchical phrase-based translation model also improved results in this translation scenario [16].

This setting and language models produced the score of BLEU equal to 19,81. Lastly we used all parallel data we were able to obtain. We adapted it using Modified Moore Levis Filtering [8]. From our experiments we conducted that best results are obtained when sampling about 150,000 bi-sentences from in-domain corpora and by using filtering after the word alignment. The ratio of data to be kept was set to 0,9 obtaining our best score equal to 22,76.

Table 6: Polish-to-English translation

System	Year	BLEU	NIST	TER	METEOR
BASE	2010	16,70	5,70	67,83	49,31
BEST	2010	23,74	6,25	54,63	57,06
BASE	2011	20,40	5,71	62,99	53,13
BEST	2011	28,00	6,61	51,02	61,23
BASE	2012	17,22	5,37	65,96	49,72
BEST	2012	23,15	5,55	56,42	56,49
BASE	2013	18,16	5,44	65,50	50,73
BEST	2013	28,62	6,71	57,10	58,48
BASE	2014	14,71	4,93	68,20	47,20
BEST	2014	18,96	5,56	64,59	51,29

The experiments on our best systems were conducted with the use of the test data from years 2010-2014. These results are showed in Table 6 and Table 7, respectively, for the Polish-to-English and English-to-Polish translations. They are measured by the BLEU, NIST, TER and METEOR metrics. Note that a lower value of the TER metric is better, while the other metrics are better when their values are higher.

Table 7: English-to-Polish translation

System	Year	BLEU	NIST	TER	METEOR
BASE	2010	9,95	3,89	74,66	32,62
BEST	2010	22,76	5,83	60,23	49,18
BASE	2011	12,56	4,37	70,13	36,23
BEST	2011	29,20	6,54	55,02	51,48
BASE	2012	10,77	3,92	75,79	33,80
BEST	2012	26,33	5,93	60,88	47,85
BASE	2013	10,96	3,91	75,95	33,85
BEST	2013	26,61	5,99	59,94	48,44
BASE	2014	9,29	3,47	82,58	31,15
BEST	2014	16,59	4,48	73,66	38,85

6. Discussion & Conclusions

Several conclusions can be drawn from the experimental results presented here. Automatic and manual cleaning of the training files has some positive impact, among the variations of the experiments [5]. Obtaining and adapting additional bi-lingual and monolingual data produced the biggest influence on the translation quality itself. In each direction using OSM and adapting training and tuning parameters was necessary and it could not be simply replicated from other experiments.

What was uncommon and surprising the punctuation normalization and usage of the hierarchical phrase model improved the quality only in translation into the Polish language and had negative results in opposite direction experiments.

What is more, converting Polish surface forms of words to lemma reduces the Polish vocabulary, which should improve the English-to-Polish translation performance and opposite. The Polish to English translation typically outscores the English to Polish translation, even on the same data. It is also what we would expect in our experiments with lemma, nonetheless our initial assumptions were not confirmed in empirical tests.

Several potential opportunities for future work are of interest. Additional experiments using extended language models are warranted to determine if this improves SMT scores. We are also interested in developing some more web crawlers in order to obtain additional data that would most likely prove useful. What is more, the Wikipedia corpus we

created is still very noisy. We are currently working on cleaning it semi-automatically.

In future we intend to try clustering the training data into word classes in order to obtain smoother distributions and better generalizations. Using class-based models was shown to be useful when translating into morphologically rich languages like Polish [23]. We are also planning on using Unsupervised Transliteration Models, that proved to be quite useful in MT for translating OOV words, for disambiguation and for translating closely related languages [24]. This feature would most likely help us overcome difference in the vocabulary size, especially when translating into PL. Using a Fill-up combination technique (instead of interpolation) that is useful when the relevance of the models is known a priori: typically, when one is trained on in-domain data and the others on out-of-domain data is also in our interests [25].

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation. We would like to test such methodology on PL-EN language pair in accordance to [22].

7. Acknowledgements

This work is supported by the European Community from the European Social Fund within the Interkadra project UDA-POKL-04.01.01-00-014/10-00 and Eu-Bridge 7th FR EU project (grant agreement n°287658).

References

[1] *M. Cetollo, J. Niehues, S. Stuker, L. Bentivogli, M. Federico, "Overview of the IWSLT2012 Evaluation Campaign", in* Proceedings of the 10th International Workshop on Spoken Language Translation (IWSLT)*,* Hong Kong, China, 2012

[2] *M. Cetollo, J. Niehues, S. Stuker, L. Bentivogli, M. Federico, "Report on the 10th IWSLT Evaluation Campaign", in* Proceedings of the 10th International Workshop on Spoken Language Translation (IWSLT)*,* Heidelberg, Germany, 2013

[3] IWSLT2014 Evaluation Campaign, http://workshop2014.iwslt.org/

[4] https://sites.google.com/site/iwsltevaluation2014/mt-track

[5] K. Wołk, K. Marasek, "Polish – English Speech Statistical Machine Translation Systems for the IWSLT 2013", in *Proceedings of the 10th International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013

[6] K. Wołk, K. Marasek, „A Sentence Meaning Based Alignment Method for Parallel Text Corpora Preparation", *Advances in Intelligent Systems and Computing volume 275*, Madeira Island, Portugal, 2014, pp. 107-114

[7] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, R. Bojar, A. Constantin, E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation", in *Proceedings of the ACL 2007 Demo and Poster Sessions*, Prague, June 2007, pp. 177–180

[8] K. Wołk, K. Marasek, „Polish -English Statistical Machine Translation of Medical Texts", *New Research in Multimedia and Internet Systems*, September 2014, pp. 169-177

[9] F. Graliński, K. Jassem, M. Junczys-Dowmunt, „PSI-Toolkit: Natural language processing pipeline", *Computational Linguistics - Applications*, Heidelberg: Springer 2012, pp. 27-39

[10] O. Bojar, "Rich Morphology and What Can We Expect from Hybrid Approaches to MT", *Invited talk at International Workshop on Using Linguistic Information for Hybrid Machine Translation(LIHMT-2011)*, 2011

[11] P. Koehn, "What is a Better Translation?", *Reflections on Six Years of Running Evaluation Campaigns*, 2011

[12] J. Tiedemann, "Parallel Data, Tools and Interfaces in OPUS", in *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, 2012, pp. 2214-2218

[13] K. Wołk, K. Marasek, „Enhanced Bilingual Evaluation Understudy", *Lecture Notes on Information Theory, volume 2 number 2*, 2014, pp.191-197

[14] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation", *Proceedings of 7th Conference of the Assoc. for Machine Translation in the Americas*, Cambridge, August 2006.

[15] K. Wołk, K. Marasek, „Real-Time Statistical Speech Translation", *Advances in Intelligent Systems and Computing volume 275*, Madeira Island, Portugal, 2014, pp.107-114

[16] D. Chiang, "Hierarchical Phrase-Based Translation", *Computational Linguistics Volume 33, Number 2*, 2007

[17] K. Heafield, "KenLM: Faster and smaller language model queries", *Proceedings of Sixth Workshop on Statistical Machine Translation, Association for Computational Linguistics*, 2011

[18] M. Costa-Jussa, J. Fonollosa, "Using linear interpolation and weighted reordering hypotheses in the Moses system", Barcelona, Spain, 2010

[19] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit", *INTERSPEECH*, 2002.

[20] Q. Gao, S. Vogel, "Parallel Implementations of Word Alignment Tool", *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, June 2008, pp. 49-57

[21] N. Durrani, H. Schmid, A. Fraser, "A Joint Sequence Model with Integrated Reordering", *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, June 19-24, 2011, pp. 1045–1054,

[22] D. Bahdanau, K. Cho, Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", arXiv cs.CL 1409.0473, 2014

[23] N. Durrani, P. Koehn, H. Schmid, A. Fraser, "Investigating the Usefulness of Generalized Word Representations in SMT", *Proceedings of the 25th Annual Conference on Computational Linguistics (COLING)*, Dublin, Ireland, August, 2014

[24] N. Durrani, P. Koehn, H. Hoang, H. Sajjad, "Integrating an Unsupervised Transliteration Model into Statistical Machine Translation", *EACL2014*, Gothenburg, Sweden, 2014

[25] *A. Bisazza, N. Ruiz, M. Federico,* "Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation*", In Proceedings of IWSLT 2011*, 2011, pp. 136-143

[26] G. Berrotarán, R. Carrascosa, A. Vine, Yalign documentation, http://yalign.readthedocs.org/en/latest/

[27] G. Musso, Sequence Alignment (Needleman-Wunsch, Smith-Waterman), http://www.cs.utoronto.ca/~brudno/bcb410/lec2notes.pdf

[28] Thorsten Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", *Lecture Notes in Computer Science Volume 1398*, 1998, pp 137-142, 2005

[29] B. Hsu, J. Glass, "Interative Language Model Estimation: Efficient Data Structure & Algorithms", *In Proceedings Interspeech*, 2008

The RWTH Aachen Machine Translation Systems for IWSLT 2014

Joern Wuebker, Stephan Peitz, Andreas Guta and Hermann Ney

Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
Aachen, Germany
<surname>@cs.rwth-aachen.de

Abstract

This work describes the statistical machine translation (SMT) systems of RWTH Aachen University developed for the evaluation campaign *International Workshop on Spoken Language Translation* (IWSLT) 2014. We participated in both the MT and SLT tracks for the English→French and German→English language pairs and applied the identical training pipeline and models on both language pairs. Our state-of-the-art phrase-based baseline systems are augmented with maximum expected BLEU training for phrasal, lexical and reordering models. Further, we apply rescoring with novel recurrent neural language and translation models. The same systems are used for the SLT track, where we additionally perform punctuation prediction on the automatic transcriptions employing hierarchical phrase-based translation. We are able to improve RWTH's 2013 evaluation systems by 1.7-1.8% BLEU absolute.

1. Introduction

We describe the statistical machine translation (SMT) systems developed by RWTH Aachen University for the evaluation campaign of IWSLT 2014. We participated in the machine translation (MT) track and the spoken language translation (SLT) track for the language pairs English→French as well as German→English. A single training pipeline with identical models using a state-of-the-art phrase-based translation engine has proven highly effective on all tasks. The pipeline includes a hierarchical reordering model, word class (cluster) language models, discriminative phrase training and rescoring with novel recurrent neural language and translation models. For the spoken language translation task, the ASR output is enriched with punctuation and casing. The enrichment is performed by a hierarchical phrase-based translation system.

This paper is organized as follows. In Sections 2.1 through 2.3 we describe our translation software and baseline setups. Sections 2.4 and 2.5 provide further details about our discriminative phrase training and the recurrent neural network models, which have proven very effective in the shared task. Our experiments for each track are summarized in Section 3 and we conclude with Section 4.

2. SMT Systems

For the IWSLT 2014 evaluation campaign, RWTH utilized state-of-the-art phrase-based and hierarchical translation systems. GIZA++ [1] is employed to train word alignments. We evaluate in case-insensitive fashion[1], using the BLEU [2] and TER [3] measures.

2.1. Phrase-based Systems

Our phrase based decoder is the implementation of the *source cardinality synchronous search* (SCSS) procedure described in [4] in RWTH's open-source SMT toolkit Jane 2.3[2] [5] , which is freely available for non-commercial use. We use the standard set of models with phrase translation probabilities and lexical smoothing in both directions, word and phrase penalty, distance-based reordering model, n-gram target language models and and enhanced low frequency feature [6]. The parameter weights are optimized with MERT [7] towards the BLEU metric. Additionally, we make use of a hierarchical reordering model (HRM) [8], a high-order word class language model (wcLM) [9], a discriminative phrase training scheme (cf. Section 2.4) and rescoring with recurrent neural network language and translation models (cf. Section 2.5).

2.2. Hierarchical Phrase-based System

For our hierarchical setups, we also employed the open source translation toolkit Jane 2.3 [10]. In hierarchical phrase-based translation [11], a weighted synchronous context-free grammar is induced from parallel text. In addition to contiguous *lexical* phrases, *hierarchical* phrases with up to two gaps are extracted. The search is carried out with a parsing-based procedure. The standard models integrated into our Jane systems are: Phrase translation probabilities and lexical smoothing probabilities in both translation directions, word and phrase penalty, binary features marking hi-

[1] We find case-insensitive evaluation more consistent with human perception of translation quality.

[2] http://www-i6.informatik.rwth-aachen.de/jane/

erarchical phrases, glue rule, and rules with non-terminals at the boundaries, extended low frequency feature and an n-gram language model. We utilize the cube pruning algorithm [12] for decoding.

2.3. Backoff language models

Each translation system uses three backoff language models that are estimated with the KenLM toolkit [13] and are integrated into the decoder as separate models in the log-linear combination: A large general domain 5-gram LM, an in-domain 5-gram LM and a 7-gram word class language model (wcLM). All of them use interpolated Kneser-Ney smoothing. For the general domain LM, we first select $\frac{1}{2}$ of the English Shuffled News, and $\frac{1}{4}$ of the French Shuffled News as well as both the English and French Gigaword corpora by the cross-entropy difference criterion described in [14]. The selection is then concatenated with all available remaining monolingual data and used to build and unpruned language model. The in-domain language models are estimated on the TED data only. For the word class LM, we train 200 classes on the target side of the bilingual training data using an in-house tool similar to `mkcls`. With these class definitions, we apply the technique shown in [9] to compute the wcLM on the same data as the general-domain LM.

2.4. Maximum Expected BLEU Training

Discriminative training is a powerful method to learn a large number of features with respect to a given error metric. In this work we learn three types of features under a maximum expected BLEU objective [15]. We perform discriminative training on the TED portion of the data, which is high quality in-domain data of reasonable size. This makes training feasible while at the same time providing an implicit domain adaptation effect. Similar to [15], we generate 100-best lists on the training data which are used as training samples for a gradient based update method. A leave-one-out heuristic [16] is applied to circumvent over-fitting. Here, we follow an approach similar to [17], where each feature type is first discriminatively trained, then condensed into a single feature for the log-linear model combination and finally optimized with MERT. In a first pass, we simultaneously train phrase pair features and phrase-internal word pair features, adding two models to the log-linear combination. Afterwards we perform a second pass focusing on reordering, with the identical feature set as the HRM, resulting in an additional six models for log-linear combination: Three orientation classes (monotone, swap and discontinuous) in both directions. As the training procedure is iterative, we select the best iteration after performing MERT. In the tables in Section 3 we denote the first pass as *maxExpBleu phr+lex* and the second pass as *maxExpBleu RO*.

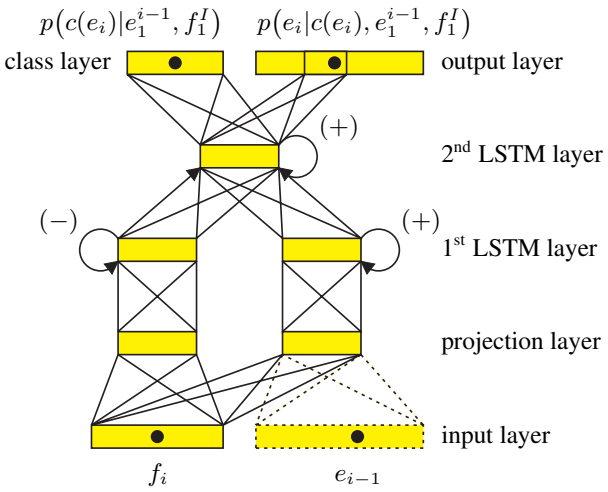


Figure 1: Architecture of the deep recurrent bidirectional translation model. By $(+)$ and $(-)$, we indicate a processing in forward and backward time directions, respectively. The inclusion of the dashed parts leads to a bidirectional *joint* model, which was not applied in this work. A single source projection matrix is used for the forward and backward branches.

2.5. Recurrent Neural Network Models

All systems apply rescoring on 1000-best lists using recurrent language and translation models. The recurrency is handled with the long short-term memory (LSTM) architecture [18] and we use a class-factored output layer for increased efficiency as described in [19]. All neural networks were trained on the TED portion of the data with 2000 word classes. In addition to the recurrent language model (RNN-LM), we apply the deep bidirectional word-based translation model (RNN-BTM) described in [20]. This requires a one-to-one word alignment, which is generated by introduction of ε tokens and using an IBM1 translation table. We apply the *bidirectional* version of the translation model, which uses both forward and backward recurrency in order to take the *full source context* into account for each translation decision. The language models are set up with 300 nodes in both the projection and the hidden LSTM layer. For the BTM, we use 200 nodes in all layers, namely the forward and backward projection layers, the first hidden layers for both forward and backward processing and the second hidden layer, which joins the output of the directional hidden layers. The architecture of the BTM network is shown in Figure 1.

3. Experimental Evaluation

3.1. English→French

For the English→French task, the word alignment was trained with GIZA++ and we applied the phrase-based decoder implemented in Jane. We used all available parallel

data for training the translation model. As backoff language models, the baseline contains a general-domain LM, an in-domain LM and a word class LM (wcLM), which are described in Section 2.3. The hierarchical reordering model (HRM) is also contained in the baseline. Experimental results are given in Table 1. By maximum expected BLEU training of phrasal and lexical features, the baseline is improved by 0.7% BLEU absolute on tst2010 and 1.5% BLEU absolute on tst2011. Including the discriminatively trained reordering model yields further gains of 0.3 and 0.1 BLEU points. The recurrent language model gives us an additional 0.7 and 0.6 BLEU points and adding the recurrent translation model, we get 0.7% and 0.2% BLEU absolute on top. The observed improvements are confirmed on the blind evaluation set tst2014, on which the scores were computed by the workshop organizers. Thus, by applying only two general and language-independent methods, our state-of-the-art baseline is improved by 2.4% BLEU on tst2010, 3.5% BLEU on tst2011 and 2.7% BLEU on tst2014. Altogether compared to last year [21] our translation performance was increased by 1.7% BLEU and 1.5% TER absolute on tst2010.

3.2. German→English

Similar to English→French, the word alignment was trained with GIZA++ and we applied the phrase-based decoder implemented in Jane. We used all available parallel data for training the translation model. The baseline contains three backoff language models, namely a general-domain LM, an in-domain LM and a word class LM as described in Section 2.3, and the hierarchical reordering model (HRM). In a preprocessing step the German source was decompounded [22] and part-of-speech-based long-range verb reordering rules [23] were applied. In addition, we tuned our system on two different development sets (dev2010 and dev2012). Since the development set from 2010 is German translated from English talks, dev2012 contains manual transcriptions from German talks. As a real test set for the manual transcription is missing, we will focus on the results (cf. Table 2) for the dev2010-tuned system in the following description. By maximum expected BLEU training of phrasal and lexical features, the baseline is improved by 1.0% BLEU absolute on tst2010 and 1.6% BLEU absolute on tst2011. Including the discriminatively trained reordering model yields further gains of 0.4 and 0.2 BLEU points. The recurrent language model gives us an additional 0.7 and 1.1 BLEU points and adding the recurrent translation model, we get 0.7% and 0.6% BLEU absolute on top. Thus, we were able to improve the state-of-the-art baseline by 2.8% BLEU on tst2010 and 3.5% BLEU on tst2011 using the same two general and language-independent methods as in the English→French task. Compared to last year [21] our translation performance was increased by 1.8% BLEU and 2.2% TER absolute on tst2010. However, we submitted the system tuned on dev2012, which contains transcribed and translated German TED-X talks and is therefore more similar to the evaluation data. The improvements are similar to the system tuned on dev2010. Unfortunatelly, they do not carry over to the blind evaluation data tst2014 in the same magnitude, where we only observe a 0.8% gain over the baseline.

3.3. Spoken Language Translation (SLT)

RWTH participated in the English→French and German→English SLT tasks. For both language pairs, we reintroduced punctuation and case information before the actual translation similar to [24]. However, we employed a hierarchical phrase-based system with a maximum of one nonterminal symbol per rule in place of a phrase-based system. A punctuation prediction system based on hierarchical translation is able to capture long-range dependencies between words and punctuation marks and is more robust for unseen word sequences. The model weights are tuned with standard MERT on 100-best lists. As optimization criterion we used F_2-Score rather than BLEU or WER. More details can be found in [25].

Since punctuation prediction and recasing were applied before the actual translation, our translation systems could be kept completely unchanged and we were able to use our final systems from the MT track directly.

4. Conclusion

RWTH participated in two MT tracks and two SLT tracks of the IWSLT 2014 evaluation campaign. The baseline systems utilize our state-of-the-art phrase-based translation decoder and we were able to improve them by discriminative phrase training (up to +1.8 BLEU) and recurrent neural network models (up to +1.9 BLEU).

For the SLT track, the ASR output was enriched with punctuation and casing information by a hierarchical translation system tuned on F_2-Score.

All presented final systems are used in the EU-Bridge system combination [26].

5. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

6. References

[1] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[2] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, July 2002, pp. 311–318.

Table 1: Results for the English→French MT task. The scores on `tst2014` were computed by the task organizers.

system	dev2010		tst2010		tst2011		tst2014	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
SCSS 2013	30.0	53.8	33.7	48.0	-	-	-	-
SCSS baseline	29.8	54.5	33.0	48.5	39.0	41.5	33.8	46.1
+maxExpBleu phr+lex	30.5	54.2	33.7	48.2	40.5	40.4	-	-
+maxExpBleu RO	30.7	54.0	34.0	47.8	40.6	40.4	35.3	44.9
+RNN-LM	31.1	53.3	34.7	47.3	41.2	39.9	-	-
+RNN-BTM	31.8	52.6	35.4	46.5	42.5	39.0	36.5	43.8

Table 2: Results for the German→English MT task. The scores on `tst2014` were computed by the task organizers.

system	dev2010		dev2012		tst2010		tst2011		tst2012		tst2014	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
SCSS 2013	34.2*	45.8*	-	-	32.3	48.1	-	-	-	-	-	-
SCSS baseline	33.8*	46.5*	24.1	62.3	31.3	49.4	36.3	44.1	31.0	49.5	-	-
+maxExpBleu phr+lex	35.2*	45.2*	24.2	61.4	32.3	48.1	37.9	42.7	32.2	48.0	-	-
+maxExpBleu RO	35.4*	45.0*	24.5	61.2	32.7	47.9	38.1	42.5	32.7	47.6	-	-
+RNN-LM	35.8*	44.0*	25.6	59.8	33.4	46.9	39.2	41.4	32.9	47.1	-	-
+ RNN-BTM	36.3*	43.2*	26.2	58.7	34.1	45.9	39.8	40.6	33.5	46.0	25.0	56.1
SCSS baseline	33.0	46.0	26.8*	58.4*	30.7	48.8	37.0	42.9	30.8	48.5	24.8	55.6
+maxExpBleu phr+lex	34.0	44.4	27.1*	57.6*	32.5	46.9	38.3	41.4	32.4	46.6	-	-
+maxExpBleu RO	33.7	44.8	27.4*	57.7*	32.3	47.1	38.3	41.5	32.7	46.7	25.2	54.8
+RNN-LM	34.2	44.4	27.7*	56.6*	32.8	46.6	38.9	41.1	32.8	46.7	-	-
+RNN-BTM	34.7	44.2	27.8*	57.2*	33.2	46.5	39.4	40.7	33.2	46.3	25.6	54.6

:

[3] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA, August 2006, pp. 223–231.

[4] R. Zens and H. Ney, "Improvements in Dynamic Programming Beam Search for Phrase-based Statistical Machine Translation," in *International Workshop on Spoken Language Translation*, Honolulu, Hawaii, Oct. 2008, pp. 195–205.

[5] J. Wuebker, M. Huck, S. Peitz, M. Nuhn, M. Freitag, J.-T. Peter, S. Mansour, and H. Ney, "Jane 2: Open source phrase-based and hierarchical statistical machine translation," in *International Conference on Computational Linguistics*, Mumbai, India, Dec. 2012, pp. 483–491.

[6] B. Chen, R. Kuhn, G. Foster, and H. Johnson, "Unpacking and transforming feature functions: New ways to smooth phrase tables," in *MT Summit XIII*, Xiamen, China, Sept. 2011, pp. 269–275.

[7] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003, pp. 160–167.

[8] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 848–856. [Online]. Available: http://dl.acm.org/citation.cfm?id=1613715.1613824

[9] J. Wuebker, S. Peitz, F. Rietig, and H. Ney, "Improving statistical machine translation with word class models," in *Conference on Empirical Methods in Natural Language Processing*, Seattle, USA, Oct. 2013, pp. 1377–1381.

[10] D. Vilar, D. Stein, M. Huck, and H. Ney, "Jane: Open source hierarchical translation, extended with reordering and lexicon models," in *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July 2010, pp. 262–270.

[11] D. Chiang, "Hierarchical Phrase-Based Translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[12] L. Huang and D. Chiang, "Forest Rescoring: Faster Decoding with Integrated Language Models," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, June 2007, pp. 144–151.

[13] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable modified Kneser-Ney language model estimation,"

in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August 2013, pp. 690–696. [Online]. Available: http://kheafield.com/professional/edinburgh/estimate_paper.pdf

[14] R. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *ACL (Short Papers)*, Uppsala, Sweden, July 2010, pp. 220–224.

[15] X. He and L. Deng, "Maximum Expected BLEU Training of Phrase and Lexicon Translation Models," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, Jeju, Republic of Korea, Jul 2012, pp. 292–301.

[16] J. Wuebker, A. Mauser, and H. Ney, "Training phrase translation models with leaving-one-out," in *Proceedings of the 48th Annual Meeting of the Assoc. for Computational Linguistics*, Uppsala, Sweden, July 2010, pp. 475–484.

[17] M. Auli, M. Galley, and J. Gao, "Large Scale Expected BLEU Training of Phrase-based Reordering Models," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct 2014.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Interspeech*, Portland, OR, USA, Sept. 2012.

[20] M. Sundermeyer, T. Alkhouli, J. Wuebker, and H. Ney, "Translation modeling with bidirectional recurrent neural networks," in *Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, Oct. 2014, pp. 14–25.

[21] J. Wuebker, S. Peitz, T. Alkhouli, J.-T. Peter, M. Feng, M. Freitag, and H. Ney, "The rwth aachen machine translation systems for iwslt 2013," in *International Workshop on Spoken Language Translation*, Heidelberg, Germany, Dec. 2013, pp. 88–93. [Online]. Available: http://workshop2013.iwslt.org/downloads/The_RWTH_Aachen_Machine_Translation_Systems_for_IWSLT_2013.pdf

[22] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *Proceedings of European Chapter of the ACL (EACL 2003)*, 2003, pp. 187–194.

[23] M. Popović and H. Ney, "POS-based Word Reorderings for Statistical Machine Translation," in *International Conference on Language Resources and Evaluation*, 2006, pp. 1278–1283.

[24] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling Punctuation Prediction as Machine Translation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, Dec. 2011.

[25] S. Peitz, M. Freitag, and H. Ney, "Better punctuation prediction with hierarchical phrase-based translation," in *International Workshop on Spoken Language Translation*, Lake Tahoe, CA, USA, Dec. 2014, to appear.

[26] M. Freitag, J. Wuebker, S. Peitz, H. Ney, M. Huck, A. Birch, N. Durrani, P. Koehn, M. Mediani, I. Slawik, J. Niehues, E. Cho, A. Waibel, N. Bertoldi, M. Cettolo, and M. Federico, "Combined spoken language translation," in *International Workshop on Spoken Language Translation*, Lake Tahoe, CA, USA, Dec. 2014, to appear.

Technical Papers

Advances in Dialectal Arabic Speech Recognition: A Study Using Twitter to Improve Egyptian ASR

Ahmed Ali, Hamdy Mubarak, Stephan Vogel

Qatar Computing Research Institute
Qatar Foundation, Doha, Qatar
{amali,hmubarak,svogel}@qf.org.qa.org

Abstract

This paper reports results in building an Egyptian Arabic speech recognition system as an example for under-resourced languages. We investigated different approaches to build the system using 10 hours for training the acoustic model, and results for both grapheme system and phoneme system using MADA. The phoneme-based system shows better results than the grapheme-based system. In this paper, we explore the use of tweets written in dialectal Arabic. Using 880K Egyptian tweets reduced the Out Of Vocabulary (OOV) rate from 15.1% to 3.2% and the WER from 59.6% to 44.7%, a relative gain 25% in WER.

1. Introduction

Arabic Automatic Speech Recognition (ASR) is a challenging task because of the lexical variety and data sparseness of the language. Arabic can be considered as one of the most morphologically complex languages [1]. With more than 300 million people speaking Arabic as a mother tongue it is the 5th most widely spoken language. Modern Standard Arabic (MSA) is the official language amongst Arabic native speakers, in fact MSA is used in formal events, such as newspaper, formal speech, and broadcast news. However, MSA is very rarely used in day-to-day communication. Nearly all the Arabic speakers use Dialectal Arabic (DA) in everyday communication [2]. DA has many differences from MSA in morphology, phonology and lexicon [3]. A significant challenge in dialectal speech recognition is diglossia, in which the written language differs considerably from the spoken vernaculars [4]. The variance among different Arabic dialects such as Egyptian, Levantine or Gulf has to be considered similar to the variance among Romance languages [5]. There are many varieties of dialectal Arabic distributed over the 22 countries in the Arabic world, often several variants of the Arabic language within the same country. There is also the difference between Bedouin and Sedentary speech, which runs across all Arabic countries. However, in natural language processing, researchers have aggregated dialectal Arabic into five regional language groups: Egyptian, Maghrebi, Gulf (Arabian Peninsula), Iraqi, and Levantine [2][6].

A recent study [7] demonstrated that the use of the online User Generated Content (UGC) can help to improve the speech recognition by an average of 12.5% for the broadcast domain in French. This result on a high-resourced language like French motivates us to consider a similar approach for Egyptian dialectal Arabic, which has to be considered a low-resource language. In this paper, we report results for Egyptian Speech Recognition using limited speech data of 10 hours for training and 1.25 hours for development and testing. There has been recent interest in Egyptian speech recognition by [8][9][10]. This paper however differs from previous work by:

1. Investigating the best practices for writing Egyptian orthography, conducting experiments on both Acoustic Model (AM) and Language Model (LM), and releasing augmented Conventional Orthography for Dialectal Arabic [11] CODA guidelines for transcribing Egyptian speech.
2. Improving the dialectal Arabic speech recognition, and showing significant reduction in the word error rate using micro blog data, particularly tweets.
3. Comparing the dialectal tweet collection and the approach being used in classifying the tweets per country.

In addition, we release a tri-gram Egyptian language model, as well Egyptian lexicon that has less than 4% OOV on the test set.

2. Dialectal Arabic

Dialectal Arabic (DA) refers to the spoken language used for daily communication in Arab countries. There are considerable geographical distinctions between DAs within countries, across country borders, and even between cities and villages as shown in Figure 1[1].

Recent research [12][2][13] is based on a coarser classification of Arabic dialects into five groups namely: Egyptian (EGY), Gulf (GLF), Maghrebi (MGR), Levantine (LEV), and Iraqi (IRQ). Other dialects are classified as OTHER (see Figure 2). Zaidan [20] mentioned that this is one possible breakdown but it is relatively coarse and can be further divided into more dialect groups, especially in large regions such as the Maghreb.

[1] http://en.wikipedia.org/wiki/Arabic_dialects

Proceedings of the 11th International Workshop on Spoken Language Translation
Lake Tahoe, December 4th and 5th, 2014

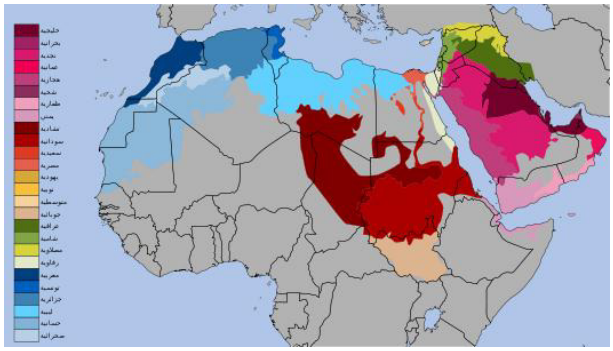


Figure 1: *Different Arabic Dialects in the Arab World.*

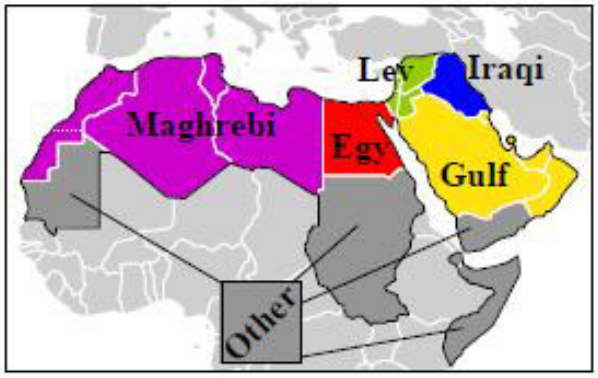


Figure 2: *Major Arabic Dialect Groups.*

3. Speech Data

3.1. Data Collection

The speech data used for this paper has been collected from Aljazeera Arabic channels, using two setups: satellite recording and internet video streaming from the Aljazeera.net website. The speech is recorded using 16 khz sampling rate. We looked at signals from both the satellite feed and online streaming, and the difference in quality is rather small and does not change anything in the quality of the audio as far as speech recognition is concerned.

A database of 200 hours has been collected over a period of six months in 2013 using the aforementioned setup. This data has been manually segmented to avoid speaker overlap, and avoid any non-speech parts such as music and background noise. These segments have a wide range of durations, from 3 seconds to 180 seconds. Speech segment were then classified as either Egyptian, Levantine, Maghrebi, Gulf, or MSA.

For the experiments described in this paper we used 12.5 hours of speech data classified to be in the Egyptian dialect, which was split into three subsets; training 10 hours, test set and development set 1.25 hours each. More details about the data are provided in Table 1.
We report the WER in this paper for both test set and development set; the first number is always for the test set and second number for the development set.

Table 1: *Speech Training Data Details.*

Duration	train(10h)	test(1.25h)	dev(1.25h)
#sentences	1385	147	176
#words	80K	9700	9809

3.2. Speech Transcription

As DA has no standard orthography or generally accepted writing convention, we investigated two approaches for manually transcribing Egyptian Speech data:
1) Verbatim transcription: The transcription is a faithful rendering of the speech without paying attention to language rules. E.g. the person name شفيق, $fyq[2] is typically pronounced by Egyptian native speakers as شفيء $fy, replacing the plosive /k/ in this context with a glottal stop hamza /A/. In this writing convention, the word will then be written as it has been pronounced, i.e. as $fy.
2) CODA-S (Augmented Coda for Speech Transcription): This transcription follows the CODA transcription guidelines [11], however, with some enhancements described below to address the needs for transcribing speech. In this case the transcription follows the language rules rather than the variant pronunciation.

CODA is mainly a framework for writing dialectal Arabic, but when working with transcribers it became apparent that some details were underspecified. We therefore augmented the CODA guidelines to make the rules clearer to the transcribers. We share these modified transcription guidelines and make them available `http://alt.qcri.org/resources/speech/Egyptian/EgyptianTranscription_CODA.pdf`

Here are some of the added explanations to the transcription guidelines. The shared document summarizes all cases by describing the case and providing samples of different writings in addition to the correct writing, as shown in Table 2, which shows one of the cases: Prefixes for future tense ("ح H" and "ه h") that are attached to present verbs, should be kept as they are without splitting from verbs

Table 2: *Examples of augmented CODA Guidelines.*

Various Writings	Correct Writing
حيبقي Hybqy, حيبقَا HybqA	حيبقَى HybqY
هَا يبقَى hA ybqY, هيبقي hybqy	هيبقَى hybqY

More rules have been added to cover cases not mentioned in the original CODA framework:

[2]Buckwalter encoding is used throughout the paper.

Split letter "ع E" that represents the preposition "علَى ElY" when concatenated to a noun. Ex: عَالأَرض EAl>rD → ع الأَرض E Al>rD.

Correct the suffix "و w", which is written instead of suffix "ه h". Ex: منو mnw → منه mnh, and عندو Endw → عنده Endh.

Restore "أ >" at the beginning of a present verb when the verb is prefixed by "ب b". Ex: بهزر bhzr → بأَهزر b>hzr.

Replace suffix "يَا yA" which indicates possession for the first person with suffix "يّ y ". Ex: فيَا fyA → فيّ fy .

Split negation article "مَا mA" in all cases. Ex: مفيش mfy$ → مَا فيش mA fy$, مَاعملتهَاش mAEmlthA$ → مَا عملتهَاش mA EmlthA$.

The guidelines also contain a new rule for punctuation marks and tags for hesitations or incomplete words, which is very important in speech transcription task. Ex: طيب ازَاي احلَى حَاجة يقولّك يد خفية Tyb Azay AHlY HAjp yqwllk yd xfyp → Tyb طيب ازَاي؟ أَحلَى حَاجة يقولّك يد خفية! <zAy? >HlY HAjp yqwl lk yd xfyp!

Finally, we added a long list of common words with different writings and the correct writing for each word. Ex: كدة kdp, كدَا kdA → كده kdh, and برضة brDp, بردو brdw → برضه brDh.

4. Dialectal Tweet Corpus

According to Twitter, the estimated number of Arabic microblogs is in excess of 15 million per day (private communication). To build a dialectal tweet corpus a multi-step procedure was used: 1) Arabic tweets were extracted by issuing the query lang:ar against the Twitter API[3].

2) Each tweet was classified as dialectal or not dialectal.

3) Dialectal tweets were mapped, if possible, to a country. If such a mapping was possible, the tweet was classified as being written in the dialect associated with that country according to Figure 2.

In more detail: To perform step 2, dialectal words were extracted from the Arabic Online Commentary Dataset (AOCD) described in [20]. Examples of words used in dialects: دي dy, عشَان E$An, هيك hyk, ايش Ay$, اكو Ako, شنو $nw, وَاش wA$ etc. As shown in [14], many of these dialectal words are used in more than one dialect. I.e. these words do not map a tweet uniquely to a dialect. For example the word "دي dy" is used in Egypt and Sudan, and the word "عشَان E$An" is used in Egypt and Arab Gulf countries etc.

If a tweet has at least one dialectal word, it was considered as dialectal tweet.

In step 3 user location in his/her profile was harvested and an attempt was made to identify the country with the aid of the GeoNames[4] geographical database. For examples: dialectal tweets with user locations like الريَاض AlryAD, Riyadh, KSA, الحجَاز AlHjAAz are mapped to Saudi Arabia and thereby to Gulf Arabic.

Applying the 3 filtering steps a corpus of size 6.5M tweets was collected during March 2014. The classification resulted in the following distribution: 3.99M tweets for Saudi Arabia (SA) (or 61% of the corpus size), 880K tweets for Egypt (EG) (13%), 707K tweets for Kuwait (KW) (11%), 302K for Arab Emirates (AE) (5%), etc. Tweets distribution is shown in Figure 3.

Using CrowdFlower[5] and 3 judges from Egypt we evaluated the accuracy for the automatic classification. Using 6,000 tweets classified as Egyptian, the achieved precision was 94%.

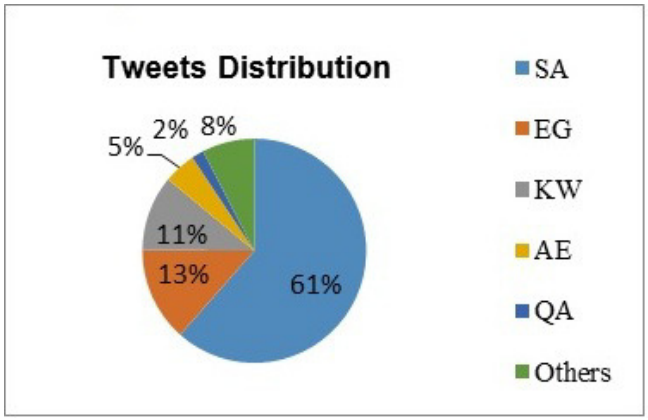


Figure 3: *Dialectal Tweets Distribution Percentages.*

5. Speech Recognition

This section describes the details of the speech recognition system, esp. the acoustic model training and the language models used in the experiments.

5.1. Language Modeling

Following [7] we wanted to test the impact of using tweets when building the language model for the speech recognition system. This leads to a number of questions: Is it better to use all dialectal tweets across the different dialects or is it better to use only the tweets in the matching dialect? How much do we gain by using more data? Does normalizing the tweets matter?

[3] http://dev.twitter.com/

[4] http://www.geonames.org/

[5] https://crowdflower.com

5.1.1. *Training Language Models*

We build standard trigram LMs with Kneser-Ney smoothing using SRI LM toolkit [18]. For interpolating LMs, the development set was used to tune the weight for the linear interpolation. In such cases we report only test set results, whereas in other cases we report numbers for both development and test set.

5.1.2. *Type/Token Ratios*

To answer the questions raised above we analyzed and compared several copora:

1) Speech data in verbatim format.
2) Speech data in CODA-S format.
3) Egyptian tweets without normalization.
4) Egyptian tweets with normalization, where we use the normalization method described in [19].
5) MSA sample, collected from the last 5 years of Aljazeera website.

One concern in statistical modeling is always data sparseness. When building language models data sparseness can be expressed in terms of type/token ratio. The higher the type/token ratio, the sparser the data becomes for LM training.

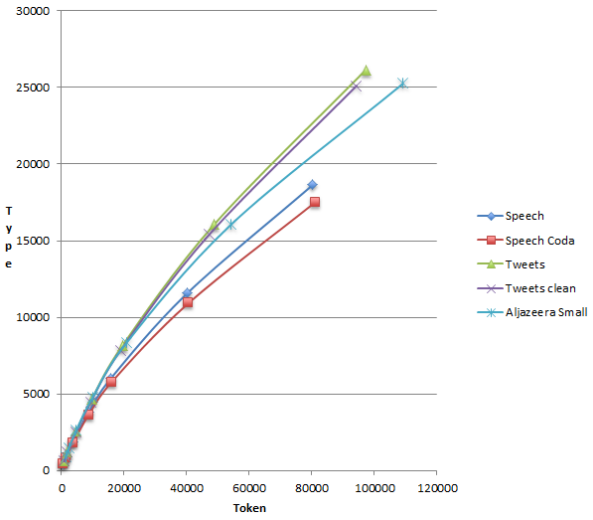


Figure 4: *Type Token Ratios for Various Text Samples.*

Figure 4 compares the type/token ratios across all the aforementioned corpora. This diagram shows how the vocabulary (number of types) grows as the corpus (number of tokens) grows. A number of observations can be made from this graph:

1) As expected, speech data shows a slower vocabulary growth compared to text data.
2) Using the CODA-S transcriptions reduces the type/token ratio, which should be benefitial for the performance of the speech recognition system.
3) The tweet corpus shows a higher type/token ratio than both speech and web-text corpora. This was not necessarily expected and could indicate that variants in writing are a major factor in dialectal tweets.
4) Normaling tweets had only a minimal effect in improving the type/token ratio. Perhaps this could be improved with a tweet-optimized normalizer rather than the simple one [19] used here.

5.1.3. *Out of Vocabulary Rates and Perplexities*

In the next step we investigated the benefit of going towards larger vocabularies, also comparing Egyptian-only tweets (TweetsEGY) versus all dialectal tweets (TweetsALL). In this comparison we looked at OOV rates and at LM perplexities, which are based on interpolated LMs: one LM build on the speech corpus in CODA-S format, one LM build on a subset of the tweets.

As shown in Table 3 the Egyptian tweets have better results on the Egyptian test set. While the gains are not very big the difference actually grows with larger vocabulary sizes. For example the drop in OOV from TweetsAll to TweetsEGY is 15% for the 30K corpus, yet 20% for the 400k corpus. The perplexity drop is even more pronounced, going from 1.3% on the 30k corpus to 5.1% on the 400k corpus.

Table 3: *Compare tweetsEGY to tweetsAll LM.*

Data	Vocab	Perplexity	OOV
ALL	30K	1096	11.6%
EGY		1082	10.2%
ALL	50K	1269	9.4%
EGY		1242	8.4%
ALL	100K	1549	7.2%
EGY		1547	6%
ALL	200K	1891	5.3%
EGY		1834	4.2%
ALL	400K	2157	4.0%
EGY		2047	3.2%

Numbers reported in Table 3 are for the test data only, as we used it to tune the LM interpolation for the training data LM and tweet data LM.

The 400K interpolated LM and the corresponding lexicon have been released on XXX web portal [6] .

5.2. Acoustic Modeling

Our acoustic models are trained with the standard 13-dimensional Cepstral Mean-Variance Normalized (CMVN)

[6] Hidden for annonymous reviewing

Mel-Frequency Cesptral Coefficients (MFCC) features without energy, and its first and second derivatives. For each frame, we also include its neighboring +/-4 frames and apply Linear Discriminative Analysis (LDA) transformation to project the concatenated frames to 40 dimensions, followed by Maximum Likelihood Linear Transform (MLLT). We use this setting of feature extraction for all models trained in our system. Speaker adaptation is also applied with feature-space Maximum Likelihood Linear Regression (fMLLR).

Our system includes all conventional models supported by KALDI [15]: diagonal Gaussian Mixture Models (GMM), subspace GMM (SGMM) and Deep Neural Network (DNN) models. Training techniques including discriminative training such as boosted Maximum Mutual Information (bMMI), Minimum Phone Error (MPE), and Sequential Training for DNN are also employed to obtain the best number.

These models are all standard 3-states context-dependent triphone models. The GMM-HMM model has about 9K Gaussians for 1.8K states; the SGMM-HMM model has 4.5K states and 40K total substates.

We studied two ways of modeling the speech:
1) grapheme-based modeling, where each character represents a model. In this system we have 36 speech models plus one model for silence. The 36 models represent the 36 unique characters, which appear in our speech training data.
2) We also studied a phoneme-based system, where we preprocessed the training text using the Morphological Analysis and Disambiguation for Arabic (MADA) toolkit [16], which has been used to build a vowelized dictionary. A rule-based vowelized to phonetized (V2P) mapping was then used to generate the final lexicon. The phoneme system has 36 phones: 35 speech phonemes and one phoneme for silence.

It is worth mentioning that MADA was developed for MSA and therefore may not the best tool for pre-processing dialectal Arabic. We learnt about MADAMIRA, which merges MADA [16] and AMIRA [17]. This tool provides linguistic information such as tokenization, diacritization, and part-of-speech tagging for each Arabic word received in corpus, which supports Egyptian text. However due to license restrictions, we were unable to use it in our experiments.

Table 4: *Comparing grapheme-based and phoneme-based systems, both with CODA-S transcriptions.*

Train data LM	Grapheme	Phoneme
1st pass WER	62.47%	51.27%
	68.41%	58.14%
2nd pass WER	59.63%	47.73%
	64.68%	53.73%

Table 4 shows that the phoneme system outperforms the grapheme system substantially with 20% relative reduction in WER. One reason behind this gain is that in Arabic the correspondence between phoneme and grapheme is weak. due to the short vowels, whic are not written. Consequently, mapping each grapheme as a unit will fall short to model in the GMM the different variants occuring in the training data. Also, the grapheme system needs more contexts to disambiguate between phonemes.

Although this is a nice reduction in WER, the range of the error is still high, which is not a surprise given the high OOV rate and perplexity. Which raises the question: is it possible to use the Egyptian tweets to build better language model to improve the dialect speech recognition? This will be addresssed in the experiments described in the next section.

6. Experiments

6.1. CODA-S and Verbatim Comparison

In an attempt to depict which approach is more appropriate to use for transcribing the Egyptian speech, we used two techniques to evaluate best approach by reporting OOV, Perplexity (PP) and ASR system and report WER.
a- Evaluating using Language Model only (LM) the test and dev set with the collected Egyptian tweets, and report OOV and PP. We used Egyptian tweets to build trigram LM, more details about Egyptian tweets in section 4, and LM in section 5. We report PP and OOV for both CODA-S and verbatim transcription convention, and as shown in Table 5. The first value refers to test set and the second to dev set. CODA-S is getting better results in both PP and OOV.

Table 5: *PP & OOV for CODA-S and Verbatim. (Type: 395K words, Tokens: 9.5M words)*

	Verbatim	CODA-S
PP	6729	5837
	6978	6031
OOV	6.8%	4.7%
	6.3%	4.6%

b- Building Grapheme based speech recognition, and report WER.
For the speech, we investigate WER at different stages of the Acoustics Model (AM) process, however, we report only the WER at the very last stage which is Deep Neural Network DNN with Minimum Phoneme Error MPE. We report the WER at first pass and the second pass, again the first value refers to test set and the second to dev set. More details about the speech recognition system are covered in section 5.

Table 4 shows that the number of words in the verbatim transcription is 80.4K words, while the total number of words in the CODA-S transcription is 81K words. Although there is a small increase in the amount of words, there is a decrease in the vocabulary size from 18.6K words in verbatim

text to 17.5K in the CODA-S text, which represents nearly 6% reduction. This is due to more consistency in writing the text which consequently reduces the sparseness in the text. It is worth mentioning that the WER comparison may not be fair measure by itself as it is impacted by Acoustic Modeling AM as well as Language Modeling LM. Having said that, in this setup we used grapheme based AM approach in both systems to be consistent with AM, and reduce the acoustic influence on the conclusion. Also, best WER does not necessary mean the best orthographic representations. But, authors found WER could be an extra measure to consider. It is clear from Table 4 that WER, PP, and OOV in the CODA-S format are consistently outperforming the verbatim transcription, which was a go-ahead signal for us to use the CODA-S as baseline for all our further experiment.

Table 6: *WER for CODA-S and verbatim*

	Verbatim	CODA-S
Token	80.4K	81K
Type	18.6K	17.5K
1st pass WER	64.78%	62.47%
	69.77%	68.41%
2nd pass WER	61.20%	59.63%
	65.98%	64.68%
Perplexity	957	862
	976	855
OOV	16.7%	15.1%
	16.6%	15.4%

6.2. Grapheme versus Phoneme based System

At this stage, we see the best AM system is the phoneme system and the best LM is the 400K vocabulary for the interpolated LM. So, the next step is to use an interpolated LM with the phoneme system and expect that the gain in both LM and AM will propagate to the final system. One challenge in doing so is that we have to pre-process tweetsEG by MADA to generate a lexicon for the phoneme system. However, as already mentioned, MADA is not the best tool to vowelize Egyptian dialectal Arabic. So, we compared both systems in Table 4 with the 400K interpolated LM and get the WER as shown in Table 7.

Table 7: *Compare Grapheme and Phoneme CODA-S Systems Using 400K Interpolated LM.*

Tweet interpolated LM	Grapheme	Phoneme
1st pass WER	47.31%	56.22%
	54.26%	62%
2nd pass WER	44.71%	52.73%
	50.62%	58.60%

We see the LM helped the Grapheme system substantially and reduced the WER by more than 25% relative in test set (from 59.63% shown in Table 4 to 44.71% shown in Table 7), and more than 21% relative reduction in development set (from 64.68% shown in Table 4 to 50.62% shown in Table 7) in the development set. In the phoneme system this gain from the tweets LM has not only vanished, but we get an increase in error rate by 10% and 9% relative in test set and development set. At this time we assume that this increase in WER stems from the fact that we do not have access to a reasonable Egyptian vowelizer, nor a nice tool that can convert dialectal Egyptian tweets into the CODA-S format.

6.3. TweetsEG versus TweetsAll

We have also investigated the importance of doing dialect detection for the tweets, and compared the WER using the Egyptian tweets versus random selection for any Arabic tweets. We see in Table 8, the dialect identification does give us some mileage. We can see a difference in WER of about 3 points absolute across both decoder passes. We report the WER on the test set only as the development set has been used to tune lambda for the linear interpolation.

Table 8: *Compare tweetsEG WER versus tweetsAll.*

Interpolated LM	Grapheme EG	Grapheme All
1st pass WER	47.31%	50.3%
2nd pass WER	44.71%	47.2%

7. Conclusion

Dialectal Arabic speech recognition is a challenging task when analyzing the available resources. In this paper, we report significant reduction in WER by approaching different aspects of the challenge: we standardize augmented CODA guidelines for transcribing Egyptian speech to reduce the impact of diglossia. We used tweets for improved vocabulary coverage and significantly reduced WER. Using specifically tweets classified as being written in the Egyptian dialect gave lower WER than using tweets across all dialects. We released the language model as well as the lexicon used in this paper. In future work, we plan to work on better dialectal vowelizer to be able to generate lexicons for different dialects. We will also investigate how to convert tweets into the CODA-S format automatically. Given the benefit of being dialect specific, we will analyze tweets that are not mapped to countries, and study using tweet location in addition to user location to enhance mapping accuracy, also enrich the dialectal words list and assign each dialectal word to a country or a set of countries.

8. References

[1] Diehl, F. et al., "Morphological decomposition in Arabic ASR systems", Computer Speech & Language, 26(4), pp.229243, 2012.

[2] Cotterell, R. & Callison-Burch, C., "A Multi-Dialect, Multi-Genre Corpus of Informal Written Arabic", The 9th edition of the Language Resources and Evaluation Conference, Reykjavik, Iceland: European Language Resources Association, 2014.

[3] Habash, N., Eskander, R. & Hawwari, A., "A Morphological Analyzer for Egyptian Arabic", pp.19, 2012.

[4] Elmahdy, M., Hasegawa-Johnson, M. & Mustafawi, E., "Development of a TV Broadcast Speech Recognition System for Qatari Arabic", The 9th edition of the Language Resources and Evaluation Conference (LREC), Reykjavik, Iceland, 2014.

[5] Holes, C., "Modern Arabic: Structures, Functions, and Varieties", 2004.

[6] Al-Sabbagh, R. & Girju, R., "YADAC: Yet another Dialectal Arabic Corpus", LREC, pp. 28822889, 2012.

[7] Schlippe, T. et al., "Unsupervised language model adaptation for automatic speech recognition of broadcast news using web 2.0", F. Bimbot et al., eds. INTERSPEECH, ISCA, pp. 26982702, 2013.

[8] Biadsy, F., Moreno, P.J. & Jansche, M., "Googles Cross-Dialect Arabic Voice Search", IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012), pp. 44414444, 2012.

[9] Al-Shareef, S. & Hain, T., "CRF-based Diacritisation of Colloquial Arabic for Automatic Speech Recognition", INTERSPEECH, 2012.

[10] Mousa, A.E. et al., "Morpheme-Based Feature-Rich Language Models Using Deep Neural Networks for LVCSR of Egyptian Arabic Human Language Technology and Pattern Recognition", Computer Science Department IBM T . J . Watson Research Center, Yorktown Heights, NY 10598, pp.84358439, 2013.

[11] Habash, N., Diab, M.T. & Rambow, O., "Conventional Orthography for Dialectal Arabic", LREC, pp. 711718, 2012.

[12] Zbib, R. et al., "Machine Translation of Arabic Dialects", In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 12, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 4959, 2012.

[13] Bahari, M.H. et al., "Non-negative factor analysis for GMM weight adaptation", IEEE Transactions on Audio Speech and Language Processing, 2014.

[14] Mubarak, H. and Darwish K., "Using Twitter to Collect a Multi-Dialectal Corpus of Arabic", Arabic NLP Workshop, EMNLP-2014, 2014.

[15] Povey, D. et al., "The Kaldi Speech Recognition Toolkit", IEEE Signal Processing Society, 2011.

[16] Habash, N., Rambow, O. & Roth, R., "MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization", In Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt. pp. 102109, 2009.

[17] Diab, M., "Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking", 2nd International Conference on Arabic Language Resources and Tools, 2009.

[18] Stolcke, A. & others, "SRILM-an extensible language modeling toolkit", INTERSPEECH, 2002.

[19] Darwish, K., Magdy, W. & Mourad, A., "Language processing for Arabic microblog retrieval", In Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 24272430, 2012.

[20] Zaidan, O.F. & Callison-Burch, C., "Arabic Dialect Identification", Computational Linguistics, 40(1), pp.171202, 2014.

[21] Ali, A. et al., "A Complete KALDI Recipe for Building Arabic Speech Recognition Systems", Spoken Language Technology Workshop (SLT), IEEE, 2014.

Towards Simultaneous Interpreting: The Timing of Incremental Machine Translation and Speech Synthesis

Timo Baumann[1], *Srinivas Bangalore*[2], *Julia Hirschberg*[3]

[1]Natural Language Systems Division, Department of Informatics, Universität Hamburg, Germany
[2]AT&T Labs – Research, Bedminster, USA
[3]Department of Computer Science, Columbia University, USA
mail@timobaumann.de, srini@research.att.com, julia@cs.columbia.edu

Abstract

In simultaneous interpreting, human experts incrementally construct and extend partial hypotheses about the source speaker's message, and start to verbalize a corresponding message in the target language, based on a partial translation – which may have to be corrected occasionally. They commence the target utterance in the hope that they will be able to finish understanding the source speaker's message and determine its translation in time for the unfolding delivery. Of course, both incremental understanding and translation by humans can be garden-pathed, although experts are able to optimize their delivery so as to balance the goals of minimal latency, translation quality and high speech fluency with few corrections. We investigate the temporal properties of both translation input and output to evaluate the tradeoff between low latency and translation quality. In addition, we estimate the improvements that can be gained with a tempo-elastic speech synthesizer.

1. Introduction

Today's speech-to-speech translation solutions are a long way from transparent and ubiquitous *universal translators* as envisioned in science fiction literature (e. g. [1]), for a multitude of reasons. One of the shortcomings is translation latency, which in speech can be described as the latency between when a concept can be grasped from listening to the source utterance and producing it as part of the target utterance. For swift and seamless communication across language barriers, low translation latency is key.

Incremental processing [2] is a technical means to implement interactive speech processing systems for online speech recognition [3], [4], [5], language understanding and generation [6], for speech synthesis [7]. Incremental processing has also been successfully applied to speech-to-speech translation (e. g. [8]), where it helps to bring down processing latency in an integrated system.

An important aspect of incremental processing (and hence, incremental translation) is the *granularity* at which material is being added. A fine granularity of processing is a precondition to low latency, as smaller units can more quickly be passed on to a next module. Previous work on incremental translation has focused on phrasing (based on intonation and somewhat related to meaning units) for translation [9], as phrases can easily be passed on to speech synthesis as one unit. Recently, incremental speech synthesis is progressing well at a word-by-word granularity, if some additional boundary and finality information is provided [10], [11].

In building language processing systems, joint analysis and optimization across module boundaries often greatly improves performance. The combination of speech recognition with understanding (e. g. [12]) or translation (e. g. [13]) is quite common, but this is less often done for the output side. (One notable exception is joint optimization of natural language generation and TTS [14], however not in an incremental setting.)

In this paper, we analyze the timing properties of source and target speech in an incremental machine translation setting in order to evaluate the improvements possible when combining word-by-word incremental machine translation with speech synthesis, particularly with respect to delivery latency. We do not yet actually employ fully incremental synthesis but focus our analysis on the advantages of such a synthesis technique in this contribution.

The remainder of this paper is structured as follows: in Section 2, we describe the interplay of incremental translation and the temporal unfolding of source and target speech based on an example and describe the basic strategies and evaluation metrics used in the study. In Section 3, we describe our corpus and experiment setup and present and discuss results for our basic strategies in Section 4. In Section 5, we look at advanced delivery timing that makes use of the flexibility that is made possible by incremental, just-in-time *tempo-elastic* speech synthesis. We summarize and conclude our work in Section 6 and outline future work in Section 7.

2. Timing Aspects of Simultaneous Interpreting

In a perfect world, a translator in transparent simultaneous interpreting will be able to come up with a perfect partial translation as soon as the corresponding source language word has

```
SRC: The | captain | waved | me | over | . |
TG1:       der/die*
TG2:              der Kapitän
TG3:              Der Kapitän winkte
TG4:              Der Kapitän winkte mich
TG5:              Der Kapitän winkte mich über*
TG6:              Der Kapitän winkte mich zu sich.
```

Figure 1: Depiction of successive incremental translation results (TGn) as words of the source utterance (SRC) are being processed. Wrongly translated words are marked by an asterisk(*). The challenge: given a (tokenized) input utterance, output should ideally commence immediately when correct translation results become available (but not before). Both source and target delivery durations must be taken into account.

been spoken by the source speaker.[1] Even in this case, the speech output component for incremental translation should consider when to start speaking rather than starting to speak immediately, as words in the target language may have a different duration than words in the source language; thus, the system could run out of words to speak, which would result in unnatural intermittent pauses during the utterance. Consider the example in Figure 1: here, even if the initial article is correctly translated to German "der", speech delivery should not commence immediately to avoid unnatural pauses if the next source language word might take longer to be uttered by the source speaker.[2] In Figure 1, translation output is purposefully aligned to show when respective words should ideally be delivered by synthesis in order to result in continuous speech output with minimal latency.

In an imperfect world, incremental translation will sometimes produce output that must later be revised (these words are marked with an asterisk in the figure; as luck has it, Google MT translates "the" to German "die", the female and plural form of the definite article, which turns out to be wrong in the example). Of course, a simultaneous translator should avoid speaking translations that later turn out to be wrong. Instead, it should speak with a high-enough latency to avoid short-range mistakes such as the ones shown in the figure.

Notice however, that the necessary delay to accommodate differences in delivery speed and intermittent translation errors can only be determined post-hoc, after the full utterance has been consumed. This of course defeats the goal of concurrent target language delivery.

We will present an analysis of the necessary delays per utterance under various translation conditions in Section 4. However, we believe that long-enough latency to account for all possible changes in translation cannot be the sole solution.

[1] Of course, our processing could also be concerned with sub-word units. However, that case would be conceptually similar to word-by-word processing (but potentially giving better results at the cost of higher complexity); this direction will not be considered further in the present work.

[2] This problem can be somewhat reduced by hesitation and/or lengthening capabilities: "de..r Kapitän ...").

Table 1: Some key statistics of the corpus (timings as determined by TTS; English reference data as well as token durations for *de*/*es* translations).

	count	duration in seconds		
	total	mean	stddev	median
utterances	1436	5.14	3.36	4.31
phrases (as determined by TTS)	3099	2.39	1.64	1.95
tokens	26890	.276	.172	.205
de token # and durations (in s)	27800	.328	.203	.25
es token # and durations (in s)	27275	.307	.195	.233

In order to account for long-range garden-pathing in translation (in which case translation *should* actively change its mind, just like a human in this situation), simply increasing delays is not the answer.

For this reason, we propose that automatic simultaneous interpreting modules, just like human experts, must have recovery capabilities, which enable them to cope with situations in which already-delivered parts of a translation should be revoked and replaced by a different translation. Human experts use and combine various strategies to cope with the problem [15]. We experiment here with the simplest possible solution of dealing with changes: we ignore all changes to words that have already or are currently being spoken. This causes the translation performance to deteriorate, given a fixed delay (similarly to [16]), which will also be analyzed in Section 4.

Finally, one intuitively important strategy of human experts is to vary the latency between input and output by varying speech delivery tempo. We report on our initial progress in determining overall latency and reducing it in Section 5.

3. Corpus and Experiment Setup

We use the IWSLT 2011 test set of the TED talks corpus as provided by the Web Inventory of Transcribed and Translated Talks [17]. As translation quality and stability may depend to a large extent on languages, we include analyses for three language pairs: $en \rightarrow de$, $en \rightarrow es$, and $de \rightarrow en$.[3]

We tokenize the respective source material with WASTE [18], using the included models for German and English. We then feed each of the utterances to standard, per-se non-incremental translation systems in a restart-incremental fashion: first translating just the first token, then the first two, then the first three, and so on, ending with the full utterance. This results in a large processing overhead and may confuse the translation system which may consider each input as a full utterance (while we are mostly sending partial utterances) – however it is a simple and reliable way of making non-incremental processors incremental. We decided to include all non-word tokens, as they give important clues to translation systems that are not trained on spoken data and are necessary to provide comparable BLEU score results on the TED data.

[3] Notice that we use the provided datasets 'in reverse' for $de \rightarrow en$ translation, ignoring the fact that that the original source becomes the target language in this experiment.

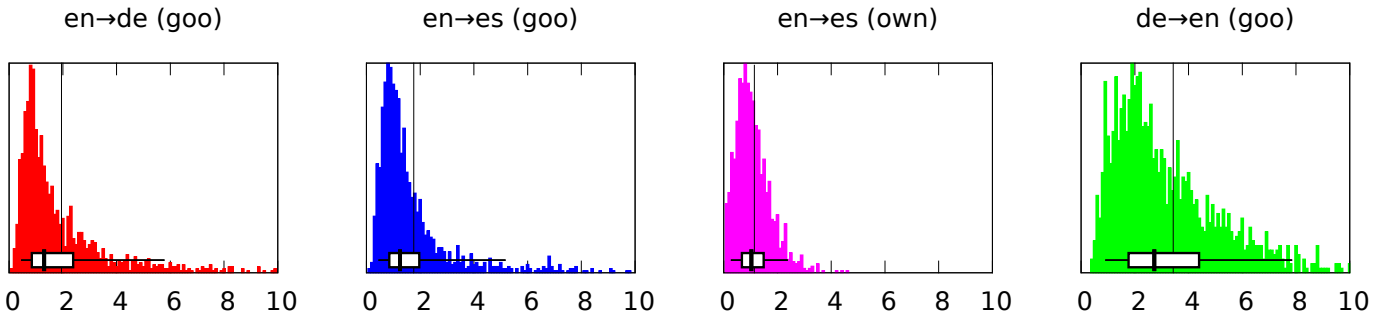


Figure 2: Histograms of per-sentence output delays (in s) that are necessary to accommodate all translation hypothesis changes.

As translation systems, we use both the Google MT web interface[4], and for Spanish, we also also use an AT&T proprietary SMT system [19].

Finally, we add word-level timing information to the source language and translation output using text-to-speech timing predictions[5] provided by MaryTTS [20] including recent additions for Spanish speech synthesis.[6] As our setup does not generate timings for some final punctuation, we use a flat duration estimate of 200 ms per punctuation in these cases. These 200 ms can be seen as the latency for end-of-utterance detection if our system were to be combined with incremental speech recognition in the future. Some key statistics about the corpus are compiled in Table 1.

As it turns out, overall German and Spanish speech duration are 23 % and 13 % higher respectively than overall English speech duration. A similar difference remains when using gold-standard German transcriptions instead of the MT output. Whether, however, this difference is due to a faster speech rate of the English voice, or due to expressive differences in the language, remains open. In any case, we have not controlled for this difference in the following experiments.

4. Evaluation of Basic Measures

For a time-aligned source sentence and its corresponding time-aligned incremental translation output that represents the final target language sentence, we find the minimum necessary delay at which the target sentence can be delivered such that the partial translation hypotheses always match the final target language sentence (i. e., the synthesis would never be triggered to start saying a word that is later replaced by a different word during incremental translation).

Using the incremental evaluation toolbox *intelida* [21], we compute the delay that is necessary in order to have all finally chosen target language words available before their scheduled delivery starts, and without intermittent interruptions from synthesis running out of words to speak. Delay histograms for all translation directions and systems are shown in Figure 2, and also indicate mean (vertical lines) as well as boxplots for median, 25/75 % (box) and 5/95 % (whiskers) quantiles. Notice that these delays are optimistic, i. e. they do not take into account translation time.

As can be seen in the histograms, the necessary delays are quite short on average, and, in particular, necessary delays for the majority of sentences are shorter than the average phrase length (cmp. Table 1), indicating that a word-level granularity (instead of phrase-level granularity as used in [9]) may be advantageous for simultaneous interpreting.

Also, we see that the histograms for the Google MT system have a very long tail with some necessary delays of over 10 seconds. On closer observation, we noticed that the Google MT system often (but not only) changes opinion when the final punctuation is added. We examined some of these sentence-final changes in detail and saw no clear tendency that they actually lead to an overall improvement of the resulting translation. In contrast, our own system, which is more strongly restricted in the sub-phrase reordering stage, results in a more normal distribution of necessary delays. This makes our own system more suitable for simultaneous interpreting, although the systems' translations and resulting BLEU scores differ, as shown below. Whether delay histograms would look more similar at equal BLEU performance levels must be left to speculation.

Finally, we notice much longer delays for $de \rightarrow en$ than for $en \rightarrow de$ translations. There may be several reasons for this: Firstly, German sentences often contain the verb late in the sentence, whereas English more stringently follows the SVO principle. As a consequence, the verb cannot be correctly translated until late in the sentence and, when it finally occurs, it may result in a change of the material that came before. Secondly, we mentioned above that our TTS generates slower speech for German (and Spanish) than for English. This phenomenon may skew the histograms in opposite directions when translating in opposite directions and may also be the cause for the longer necessary delays when translating from German. However, the histogram does not tail off as quickly

[4] `http://translate.google.com/` with the help of some PHP-based automation code.

[5] Of course, we could have extracted more precise source language timing information from TED videos, but results would likely be similar and only be available for English as source language.

[6] We thank Marcela Charfuelan for making a Spanish voice and linguistic resources available.

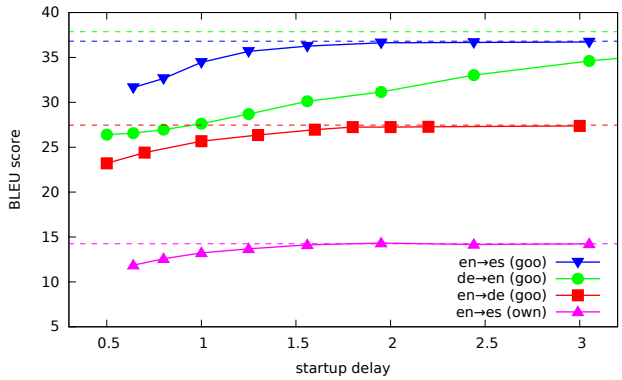


Figure 3: Performance penalty for given initial delays under all translation conditions.

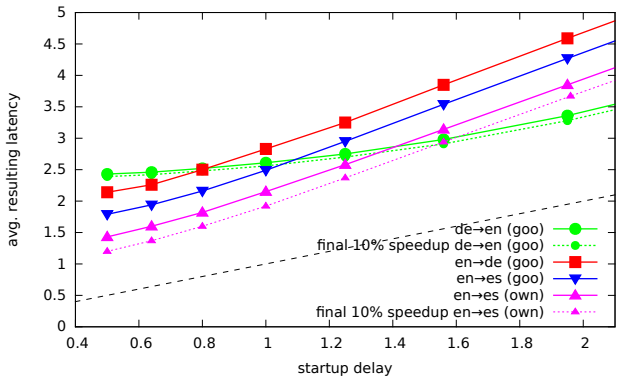


Figure 4: Resulting final latencies for given initial delays.

as it does with English as source language, which could not be explained by differing TTS tempos.

Of course, latency is just one aspect of simultaneous interpreting, the other major factor being translation quality. Beyond the non-incremental translation quality of the translation systems and corpora used, we have also implemented a very simple method for generating incremental translations under time-pressure (i.e., in simultaneous interpreting), where some words that are later overridden by a more informed translation, are already being spoken. In this case, our system simply ignores the change and re-aligns the new translation hypothesis using the Levenshtein algorithm [22].

Figure 3 shows translation performance (in terms of BLEU scores) of the different translation conditions for non-incremental (horizontal dashed lines) translation, which forms a natural upper bound for translations that are restricted in changing their hypotheses to different latency settings.

As can be seen in the figure, overall translation performance differs between translation systems, language pairs, and direction. Specifically, Google's $en \rightarrow de$ translations lags behind and differs substantially from the reverse translation direction, or $en \rightarrow es$. Our own $en \rightarrow es$ system performs poorly as compared to Google's. Our system was trained on different domain material, which may limit its performance on TED data; we plan to re-train our models in time for the final version of this paper.

Aside from translation quality, the performance penalty from limited-delay processing also differs substantially: our own system approaches its non-incremental performance rather quickly, while Google's systems require longer delays to reach their performance ceiling – although it must be noted that Google outperforms our system even with short delays.

Quite importantly, we note that $de \rightarrow en$ translation suffers most from long delays, to an extent that incremental performance is lower than $en \rightarrow es$, even though the non-incremental performance is higher, $de \rightarrow en$ only approaches non-incremental performance with a startup delay of around 4.8 seconds. We believe this property to stem from linguistic properties of German, which are not well-handled by our overly simplistic incrementalization approach.

5. Considering Speech Delivery Tempo

Words in the target language can only begin to be realized when the corresponding source language word has been completely delivered and translated. As words have different inherent durations, an incremental system may intermittently run out of material to speak, waiting for the next source language word to be completed and translated. As a consequence, the actual delay of the system as implemented is often higher than the initial startup delay, when the system has to wait for more translated material to become available. In addition, the speaking duration of the remaining words in the target language once the source utterance and translation have completed, must also be considered.

Figure 4 shows the actual resulting latency of target utterance completion after source utterance completion, at various initial delay settings. As can be seen in the figure, the resulting latency is substantially higher than the initial delay. There are two reasons for this: (a) target utterance delivery still needs to finish after the source utterance has already been completed; and (b) the delay may have to be increased intermittently, when source language delivery is too slow to sustain translation and delivery in the target language. In our experiments, we observe both phenomena, even though (a) is prevalent.

In all cases, latency is more than a second higher than the startup delay, which forms a natural lower bound for latency. We also notice that latencies increase more than initial delays for all but the $de \rightarrow en$ condition. This may be due to synthesis delivery speeds differing across languages. While $de \rightarrow en$ is impaired most by low delays (see Figure 3), it also accommodates longer delays in terms of the resulting latencies.

As mentioned previously, one goal of incremental speech synthesis is to have immediate control over delivery, and specifically, delivery timing. In Figure 4, we also plotted as dotted lines the resulting latencies if speech synthesis is sped up by 10 % starting at the word delivered after the source

language utterance has been completed. We notice a slight latency reduction across languages, of 4-5 % on average, or 100-200 ms, which may already be noticeable in applications.

We believe that somewhat higher speed-ups may be tolerable for listeners, which will lead to correspondingly larger improvements, and we plan to confirm this in listening/understanding experiments.

The second source of latency is delays that are increased during the utterance as the system runs out of target material. These additional delays can be substantial, especially for short startup delays. For example in the $en \rightarrow de$ condition, the additional average delay amounts to about 288 ms (179 ms) for a startup delay of 500 ms (respectively 700 ms).

We plan to reduce overall latency by bringing down utterance-internal delays through increasing speech tempo after the system has to intermittently pause. More generally speaking, we hope to estimate incremental translation stability (similarly to speech recognition stability [5]) and infer a flexible delay that accommodates more change at times when translation is particularly uncertain. The flexible delays will be integrated by varying delivery tempo in the incremental speech synthesis.

6. Conclusions

We have presented an analysis of incremental speech translation that takes into account speech delivery timings for both input and output. We find that, on average, conventional translation systems that are employed in a restart-incremental fashion produce their results with relatively low latencies. In particular, average delays are shorter than the average phrase, confirming our belief that word-by-word incrementality leads to better quality/latency trade-offs than phrase-by-phrase incremental systems.

In our experiments, we find that language pairs behave differently, and that German-to-English translation may be particularly difficult to perform incrementally. In addition, we find that our own system, which is quite limited in the word-reordering stage of translation, does not require as long delays and approaches its performance ceiling more quickly with limited delays – however, at the cost of overall lower performance. We plan to re-train our models with in-domain data in order to better compare our system with Google's MT.

In addition, we find that overall latency results from both the source utterance timing and its translation, and the target utterance delivery. While we have implemented a simple solution for the latter issue, we are still exploring how to deal with the former.

Finally, BLEU scores may be insufficient to judge incremental performance. An incremental translation system should strategically consider the duration of target language words in order to "gain time" or to speed up delivery, as required over the course of an utterance, while remaining easily understandable. Such word choices may hurt BLEU, as the "wrong" translation can be chosen, but improve actual system behaviour.

7. Future Work

As next steps, we will examine stability models for translations, similar to [5] for speech recognition. Our initial experiments in this direction are promising; however, they require translation internals which are not available from Google's MT. On the other hand, our own translation system is not trained on in-domain data, and hence delivers poor performance.

As we do not believe that a simultaneous interpreting system can lag behind to a degree that it "covers" all intermittent mis-translations, such a system will require an explicit recovery module that is able to rephrase and correct (perhaps using prosodic marking) already delivered material in a way that is easy to digest for the user. As such rephrasing cannot be learned from translation data, we believe this process cannot be left to the translation module alone.

Finally, we plan to validate the trade-off between translation quality and latency reduction of our system in a user study. In order to focus the study on the incremental aspects of the system, we plan to have participants fill in a multiple-choice survey about facts conveyed in the translation material. The timing of answers and their correctness will be informative regarding the two major aspects of incremental processing, latency and correctness. In addition, user changes to their answers should be useful in conveying information about the stability of the message conveyed.

8. Acknowledgements

The authors would like to thank Marcela Charfuelan for making available her MaryTTS extensions for Spanish speech synthesis, as well as the valuable feedback by the anonymous reviewers. This work is supported by a Daimler and Benz Foundation PostDoc Grant to the first author.

9. Bibliography

[1] D. Adams, *The Hitchhiker's Guide to the Galaxy*, ser. The Hitchhiker's Guide to the Galaxy. Pan Books, Oct. 1979.

[2] D. Schlangen and G. Skantze, "A General, Abstract Model of Incremental Dialogue Processing," in *Proceedings of the EACL*, Athens, Greece, 2009, pp. 710–718.

[3] T. Baumann, M. Atterer, and D. Schlangen, "Assessing and improving the performance of speech recognition for incremental systems," in *Proceedings of NAACL-HLT 2009*, Boulder, USA, 2009, pp. 380–388.

[4] E. Selfridge, I. Arizmendi, P. Heeman, and J. Williams, "Stability and accuracy in incremental speech recognition," in *Proceedings of the SIGDIAL 2011 Conference*, Portland, Oregon: Association for Computational Linguistics, Jun. 2011, pp. 110–119. [Online]. Available: `http://www.aclweb.org/anthology/W/W11/W11-2014`.

[5] I. McGraw and A. Gruenstein, "Estimating word-stability during incremental speech recognition," in *Proceedings of Interspeech*, ISCA, Portland, USA, Sep. 2012.

[6] G. Skantze and A. Hjalmarsson, "Towards incremental speech generation in dialogue systems," in *Proceedings of SIGdial*, Tokyo, Japan, Sep. 2010.

[7] T. Baumann and D. Schlangen, "INPRO_ISS: a component for just-in-time incremental speech synthesis," in *Procs. of ACL System Demonstrations*, Jeju, Korea, 2012.

[8] S. Bangalore, V. K. Rangarajan Sridhar, P. Kolan, L. Golipour, and A. Jimenez, "Real-time incremental speech-to-speech translation of dialogs," in *Proceedings of NAACL-HTL 2012*, Montréal, Canada, Jun. 2012, pp. 437–445.

[9] V. K. R. Sridhar, J. Chen, S. Bangalore, and A. Conkie, "Role of pausing in text-to-speech synthesis for simultaneous interpretation," in *Proceedings of SSW8*, 2013.

[10] T. Baumann, "Decision tree usage for incremental parametric speech synthesis," in *Proceedings of the International Conference on Audio, Speech, and Signal Processing (ICASSP 2014)*, Florence, Italy, May 2014.

[11] ——, "Partial representations improve the prosody of incremental speech synthesis," in *Proceedings of Interspeech*, 2014.

[12] A. Deoras, R. Sarikaya, G. Tur, and D. Hakkani-Tur, "Joint decoding for speech recognition and semantic tagging," Annual Conference of the International Speech Communication Association (Interspeech), 2012. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=183552.

[13] H. Ney, "Speech translation: coupling of recognition and translation," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, IEEE, vol. 1, 1999, pp. 517–520.

[14] C. Nakatsu and M. White, "Learning to say it well: reranking realizations by predicted synthesis quality," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia: Association for Computational Linguistics, 2006, pp. 1113–1120. DOI: 10.3115/1220175.1220315. [Online]. Available: http://www.aclweb.org/anthology/P06-1140.

[15] V. K. R. Sridhar, J. Chen, and S. Bangalore, "Corpus analysis of simultaneous interpretation data for improving real time speech translation.," in *INTERSPEECH*, 2013, pp. 3468–3472.

[16] H. Shimizu, G. Neubig, S. Sakti, T. Toda, and S. Nakamur, "Constructing a speech translation system using simultaneous interpretation data," in *Proceedings of the 10th International Workshop on Spoken Language Translation (IWSLT 2013)*, Heidelberg, Germany, 2013, pp. 212–218.

[17] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, 2012, pp. 261–268.

[18] B. Jurish and K.-M. Würzner, "Word and sentence tokenization with hidden markov models," *JLCL*, vol. 28, no. 2, pp. 61–83, 2013.

[19] V. kumar Rangarajan sridhar, S. Bangalore, A. Jimenez, L. Golipour, and P. Kolan, "SPECTRA: a speech-to-speech translation system in the cloud," IEEE International Conference on Emerging Signal Processing Applications, Tech. Rep., 2012.

[20] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system MARY: a tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 3, pp. 365–377, Oct. 2003, ISSN: 1572-8110. DOI: 10.1023/A:1025708916924.

[21] T. von der Malsburg, T. Baumann, and D. Schlangen, "TELIDA: A Package for Manipulation and Visualisation of Timed Linguistic Data," in *Proceedings of SigDial 2009*, London, UK, 2009.

[22] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics – Doklady*, vol. 10, no. 8, pp. 707–710, Feb. 1966.

WORD CONFIDENCE ESTIMATION FOR SPEECH TRANSLATION

L. Besacier, B. Lecouteux, N.Q. Luong, K. Hour and M. Hadjsalah

LIG, University of Grenoble, France

laurent.besacier@imag.fr

Abstract

Word Confidence Estimation (WCE) for machine translation (MT) or automatic speech recognition (ASR) consists in judging each word in the (MT or ASR) hypothesis as correct or incorrect by tagging it with an appropriate label. In the past, this task has been treated separately in ASR or MT contexts and we propose here a joint estimation of word confidence for a spoken language translation (SLT) task involving both ASR and MT. This research work is possible because we built a specific corpus which is first presented. This corpus contains 2643 speech utterances for which a quintuplet containing: ASR output (src-asr), verbatim transcript (src-ref), text translation output (tgt-mt), speech translation output (tgt-slt) and post-edition of translation (tgt-pe), is made available. The rest of the paper illustrates how such a corpus (made available to the research community) can be used for evaluating word confidence estimators in ASR, MT or SLT scenarios. WCE for SLT could help rescoring SLT output graphs, improving translators productivity (for translation of lectures or movie subtitling) or it could be useful in interactive speech-to-speech translation scenarios.

Word confidence estimation (WCE), Spoken Language Translation (SLT), Corpus, Joint features.

1. Introduction

Confidence estimation is a rather hot topic both for Automatic Speech Recognition (ASR) and for Machine Translation (MT). While ASR and MT systems produce more and more user-acceptable outputs, we still face open questions such as: are these translations/transcripts ready to be published as they are? Are they worth to be corrected or do they require retranslation/retranscription from scratch? It is undoubtedly that building a method which is capable of pointing out the correct parts as well as detecting the errors in each MT or ASR hypothesis is crucial to tackle these above issues. Also, confidence estimation can help to re-rank N-best hypotheses [1] or re-decode the search graph [2]. If we limit the concept "parts" to "words", the problem is called Word-level Confidence Estimation (WCE).

The WCE's objective is to assign each word in the MT or ASR hypothesis a confidence score (typically between 0 and 1). For error detection, this score can be binarized and then each word is tagged as correct or incorrect. In that case, a classifier which has been trained beforehand from a feature set calculates the confidence score for the output word, and then compares it with a pre-defined threshold. All words with scores that exceed this threshold are categorized in the *Good* label set; the rest belongs to the *Bad* label set. In the past, this task has been treated separately in ASR or MT contexts and we propose here a joint estimation of word confidence for a spoken language translation (SLT) task involving both ASR and MT. We believe that WCE for SLT could help improving translators productivity (for lecture or movie translation) or it could be useful in interactive speech-to-speech translation.

The remaining of this paper is the following. In section 2 we present our first contribution: a corpus (distributed to the research community) dedicated to WCE for SLT. To our knowledge, this is the first corpus that allows experimenting such a task. It contains 2643 speech utterances for which a quintuplet (containing ASR output, verbatim transcript, text translation output, speech translation output and post-edition of translation) is available. Then sections 3 and 4 present our WCE systems (as well as a quick description of related works) for ASR and MT respectively. Section 5 illustrates how our corpus can be used for evaluating word confidence estimators in a SLT scenario. Finally we conclude this paper and give some perspectives.

2. A database for WCE evaluation in spoken language translation

2.1. Starting point: an existing MT Post-edition corpus

For a French-English translation task, we used our SMT system to obtain the translation hypothesis for 10,881 source sentences taken from news corpora of the WMT (Workshop on Machine Translation) evaluation campaign (from 2006 to 2010). Post-editions were obtained from non professional translators using a crowdsourcing platform. More details on the baseline SMT system used can be found in [4] and more details on the post-edited corpus can be found in [5]. It is worth mentionning, however, that a subset (311 sentences) of these collected post-editions was assessed by a professional translator and 87.1% of post-editions were judged to improve the hypothesis

Then, the word label setting for WCE was done using TERp-A toolkit [3]. Table 1 illustrates the labels generated by TERp-A for one hypothesis and post-edition pair. Each word or phrase in the hypothesis is aligned to a word or phrase in the post-edition with different types of edit: I (in-

Reference	The	consequence	of	the	fundamentalist	movement		also	has	its importance	.
		S			S	Y	I		D	P	
Hyp After Shift	The	result	of	the	hard-line	trend	is	also		important	.

Table 1: Example of WCE label setting using TERp-A [3]

sertions), S (substitutions), T (stem matches), Y (synonym matches), and P (phrasal substitutions). The lack of a symbol indicates an exact match and will be replaced by E thereafter. We do not consider the words marked with D (deletions) since they appear only in the reference. However, later on, we will have to train binary classifiers (good/bad) so we re-categorize the obtained 6-label set into binary set: The E, T and Y belong to the *Good* (G), whereas the S, P and I belong to the *Bad* (B) category. Finally, we observed in our corpus that out of total words (train and test sets) are 85% labeled G, 15% labeled B.

From this corpus, we extract 10,000 triplets (source reference *src-ref*, machine translation output *tgt-mt* and post-edition of translation *tgt-pe*) for training our WCE (for MT) system and keep the remaining 881 triplets as a test set.

2.2. Augmenting the corpus with speech recordings and transcripts

In order to take advantage of the existing PE corpus, we decided to record the utterances of its test part to augment the corpus with speech inputs. We admit that this would have been better to capture real speech data, then transcribe it, translate and post-edit but we believe that our corpus will remain useful to study WCE for SLT, even if translating read speech is not the best practical SLT task we could imagine.

So, the test set of this corpus was recorded by French native speakers. Each of the 881 sentences was uttered by 3 speakers, leading to 2643 speech recordings. 15 speakers (9 women and 6 men) took part to the speech data collection in normal office condition. The total length of the speech corpus obtained is more than 5h since some utterances were pretty long.

Then, our French ASR system based on KALDI toolkit [6] was used to obtain the speech transcripts. The 3-gram language model was trained on the French ESTER corpus as well as French Gigaword (vocabulary size is 55k). SGMM-based acoustic models were trained using the same ESTER corpus - see details in [7].

It is important to note that automatic post-processing was needed at the output of the ASR system in order to match requirements of standard input for machine translation (we wanted our ASR outputs to match, as much as possible, our already available *src-ref* utterances). Thus, the following post-treatments were applied: number conversion (back to digit numbers), recasing (our SMT system is a true case one), re-punctuating, converting full words back to abbreviations (*kilometre* becomes *km*, *madame* becomes *Mme*, etc.) and restoring special characters (*pourcents* becomes %, *euro* becomes €). With this post-processing, the output of our ASR system, scored against the *src-ref* reference went from 29.05% WER to 26.6% WER.

This WER may appear as rather high according to the task (transcribing read news) but these news contain a lot of foreign named entities (part of the data is extracted from French newspapers dealing with european economy in many EU countries).

2.3. Obtaining labels in order to evaluate WCE for SLT

We now have a new element of our desired quintuplet: the ASR output *src-asr*. It is the noisy version of our already available verbatim transcripts called *src-ref*. This ASR output (*src-asr*) was then translated by the exact same SMT system [4] already mentionned in paragraph 2.1. This new output translation is called *tgt-slt* and it is a degraded version of *tgt-mt*.

At this point, a strong assumption we made has to be revealed: we re-used the post-editions obtained from the text translation task (called *tgt-pe*), to infer the quality (G,B) labels of our speech translation output *tgt-slt*. The word label setting for WCE is also done using TERp-A toolkit [3] between *tgt-slt* and *tgt-pe*. This assumption (as well as the fact that initial MT post-edition can be also used to infer labels of a SLT task) is reasonable regarding results (later presented in Table 4) where it is shown that there is not a huge difference between the MT and SLT performance (evaluated with BLEU). This means that if the real SLT output had been post-edited, we would have obtained very similar PE to the actual ones.

The remark above is important and this is what makes the value of this corpus. For instance, other corpora such as the TED corpus compiled by LIUM[1] contains also a quintuplet with ASR output, verbatim transcript, MT output, SLT output and target translation. But there are two main differences: first, the target translation is a manual translation of the prior subtitles so this is not a post-edition of an automatic translation (and we have no guarantee that the G/B labels extracted from this will be reliable for WCE training and testing). Secondly, in our corpus, each sentence is uttered by 3 different speakers in order to introduce a minimum of speaker variability in the test set (the consequence is that we have different ASR outputs for a single source sentence).

2.4. Final corpus statistics and web link for download

The main statistics regarding this corpus are in Table 2, where we also clarify how the WCE labels were obtained. For the test set, we now have all the data needed to evaluate WCE for 3 tasks :

[1] http://www-lium.univ-lemans.fr/fr/content/corpus-ted-lium

- **ASR**: extract G/B labels by computing WER between *src-asr* and *src-ref*,
- **MT**: extract G/B labels by computing TERp-A between *tgt-mt* and *tgt-pe*,
- **SLT**: extract G/B labels by computing TERp-A between *tgt-slt* and *tgt-pe*.

Data	# train utt	# test utt	method to obtain WCE labels
src-ref	10000	881	
src-sig		5h	speech
src-asr		881*3	wer(*src-asr*,*src-ref*)
tgt-mt	10000	881	terpa(*tgt-mt*,*tgt-pe*)
tgt-slt		881*3	terpa(*tgt-slt*,*tgt-pe*)
tgt-pe	10000	881	

Table 2: Overview of our post-edition corpus for SLT

Table 3 gives an example of quintuplet available in our corpus. One transcript (*src-hyp1*) has 1 error while the other one (*src-hyp2*) has 4. This leads to respectively 2 B labels (*tgt-slt1*) and 4 B labels (*tgt-slt2*) in the speech translation output, while *tgt-mt* has only one B label. Table 4 summarizes the MT (translation from verbatim transcripts) and SLT (translation from automatic speech transcripts) performances obtained on our corpus, as well as the distribution of good (G) and bad (B) labels inferred for both tasks. Logically, the percentage of (B) labels increases from MT to SLT task in the same conditions.

src-ref	quand	notre	cerveau	chauffe
src-hyp1	*comme*	notre	cerveau	chauffe
labels ASR	B	G	G	G
src-hyp2	*qu'*	*entre*	*serbes*	*au* chauffe
labels ASR	B	B	B	B G
tgt-mt	when	our	brains	*chauffe*
labels MT	G	G	G	B
tgt-slt1	*as*	our	brains	*chauffe*
labels SLT	B	G	G	B
tgt-slt2	*between*	*serbs*	*in*	*chauffe*
labels SLT	B	B	B	B
tgt-pe	when	our	brain	heats up

Table 3: Exemple of quintuplet with associated labels

task	ASR (WER)	MT (BLEU)	% G (good)	% B (bad)
tgt-mt	0%	36.1%	82.5%	17.5%
tgt-slt	26.6%	30.6%	65.5%	34.5%

Table 4: MT and SLT performances on our test set

This corpus is available for download on *github.com/besacier/WCE-SLT-LIG*.

3. WCE for speech transcription

3.1. Related work

Several previous works tried to propose effective confidence measures in order to detect errors on ASR outputs. Out-Of-Vocabulary (OOV) detection was introduced by [8] and extended by [9] and [10]. [9] introduced the use of word posterior probability (WPP) as a confidence measure for speech recognition. Posterior probability of a word (or a sequence) is most of the time computed using the hypothesis word graph [9] [11].

Recent approaches [12, 10] for confidence measure estimation use side-information extracted from the recognizer: normalized likelihoods (WPP), the number of competitors at the end of a word (hypothesis density), decoding process behavior, linguistics features, acoustic features (acoustic stability, duration features) and semantic features. Finally, these papers show the prominence of linguistic features.

Later, WPP score was combined with other high-level knowledge sources to improve the confidence estimation. For instance, [10] proposed an efficient method that combines various features (acoustic, linguistic, decoding and semantic features). Another work by [13] combines scores extracted from several sources: N-best features, acoustic stability, hypothesis density, duration features, language model, parsing features, WPP, etc.

3.2. WCE system used and baseline performance

In this work, we extract several types of features, which come from the ASR graph, from language model scores and from a morphosyntactic analysis. These features are listed below:

- Acoustic features : words errors probably induce acoustic distortions between the hypothesis and the best phonetic sequence. Many observations points out that word length can predict correct words and errors: we add a feature which consists of the word duration (F-dur).
- Graph features : they are extracted from the word confusion networks. When an error occurs, the search algorithm explores various alternative paths: the posterior probabilities and alternative paths can help to predict errors. We use the number of alternative (F-alt) paths in the word section, and the posterior probability (F-post).
- Linguistic features : they are based on probabilities provided by the language model (3-gram LM) used in the KALDI ASR system. We use the word itself (F-word) and the 3-gram probability (F-3g) . We also add the feature (F-back), proposed in [12] which represents the back-off level of the targeted word.
- Lexical Features: word's Part-Of-Speech (F-POS) are computed using tree-tagger for French.

We use a variant of boosting classification algorithm in order to combine features. The used implementation is Bonzaiboost[2] [14]. It implements the boosting algorithm Adaboost.MH over deeper trees.

For each word, we estimate the 7 features (F-Word; F-3g; F-back; F-alt; F-post; F-dur; F-post) previously described. The classifier is trained on BREF 120 corpus [15]. After

[2]http://bonzaiboost.gforge.inria.fr

decoding, we obtain about 1M word examples. Each word from this corpus is tagged as correct or not correct, according to the reference.

Once we have the prediction model built with all features, we apply it on the test set (3*881 sentences) and obtained the required WCE labels along with confidence probabilities. In term of F-score, our WCE system reaches the following performance: predicting *"G"* label: (**87.85%**), and predicting *"B"* label: (**37.28%**).

4. WCE for machine translation

4.1. Related work

The Workshop on Machine Translation (WMT) introduced in 2013 a WCE task for machine translation. [16, 17] employed the Conditional Random Fields (CRF) [18] model as their Machine Learning method to address the problem as a sequence labeling task. Meanwhile, [19] extended the global learning model by dynamic training with adaptive weight updates in the perceptron training algorithm. As far as prediction indicators are concerned, [19] proposed seven word feature types and found among them the "common cover links" (the links that point from the leaf node containing this word to other leaf nodes in the same subtree of the syntactic tree) the most outstanding. [16] focused only on various n-gram combinations of target words. Inheriting most of previously-recognized features, [17] integrated a number of new indicators relying on graph topology, pseudo reference, syntactic behavior (constituent label, distance to the semantic tree root) and polysemy characteristic. Optimization endeavors were also made to enhance the baseline, including classification threshold tuning, feature selection and boosting technique [17].

4.2. WCE system used and baseline performance

We employ the Conditional Random Fields [18] (CRFs) as our machine learning method, with WAPITI toolkit [20], to train the WCE model. A number of knowledge sources are employed for extracting features, in a total of 25 major feature types:

- Target Side: target word; bigram (trigram) backward sequences; number of occurrences
- Source Side: source word(s) aligned to the target word
- Alignment Context [21]: the combinations of the target (source) word and all aligned source (target) words in the window ± 2
- Word posterior probability [22]
- Pseudo-reference (Google Translate): Does the word appear in the pseudo reference or not?
- Graph topology [23]: number of alternative paths in the confusion set, maximum and minimum values of posterior probability distribution
- Language model (LM) based: length of the longest sequence of the current word and its previous ones in the target (resp. source) LM. For example, with the target word w_i: if the sequence $w_{i-2}w_{i-1}w_i$ appears in the target LM but the sequence $w_{i-3}w_{i-2}w_{i-1}w_i$ does not, the n-gram value for w_i will be 3.
- Lexical Features: word's Part-Of-Speech (POS); sequence of POS of all its aligned source words; POS bigram (trigram) backward sequences; punctuation; proper name; numerical
- Syntactic Features: null link [24]; constituent label; depth in the constituent tree
- Semantic Features: number of word senses in WordNet.

Interestingly, this feature set was also used in our English - Spanish WCE System submitted for WMT 2013 Quality Estimation shared task and obtained the best performance [23].

Once we have the prediction model, we apply it on the test set (881 sentences) and obtained the required WCE labels along with confidence probabilities. In term of F-score, our WCE system reaches very promising performance in predicting *"G"* label (**87.65%**), and acceptable for *"B"* label (**42.29%**).

5. Joint estimation of word confidence for a speech translation task

Now, if we consider WCE for a speech translation task, there is no related work available since, to our knowledge, this is the first time such a task is proposed with a corpus allowing to evaluate joint WCE features coming from both ASR and MT.

task feat. type	**WCE for ASR** ASR feat.	**WCE for MT** MT feat.	**WCE for SLT** MT feat.	**WCE for SLT** ASR feat.	**WCE for SLT** 0.5MT+0.5ASR feat.
F(G)	87.85%	87.65%	77.17%	76.41%	77.54%
F(B)	37.28%	42.29%	39.34%	38.00%	43.96%

Table 5: Summary of word confidence estimation (WCE) results obtained on our corpus with different feature sets based on ASR, MT or both. Numbers reported are F scores for Good (G) and Bad (B) labels respectively with a common decision threshold.

We first report in Table 5 the baseline results by individual WCE systems for a single ASR task and for a single MT task (second and third columns of the table - numbers correspond to the performance of the systems described in the two previous sections). Then, to illustrate how our corpus can be used for word confidence estimation in speech translation, we evaluated the performance of 3 systems (using *labels SLT* - see Table 3 - as reference to score the WCE systems):

- The first system (SLT sys. / MT feat.) is the one described in section 4 and uses only MT features. No

modification of the WCE (for MT) system is needed since the only difference is that the source sentence is *src-hyp* (ASR output) instead of *src-ref*,

- The second system (SLT sys. / ASR feat.) is the one described in section 3 and uses only ASR features. So this is predicting SLT output confidence using only ASR confidence features ! Word alignment information between *src-hyp* and *tgt-slt* is needed to project the WCE scores coming from ASR, to the SLT output (done using adequate Moses option, where the alignment information is kept in the decoding output).
- The third system (SLT sys. / MT+ASR feat.) combines the information from the two previous WCE systems. In this work, the ASR-based confidence score of the source is projected to the target SLT output and linearly combined with the MT-based confidence score (we tried different weights but only report *0.5MT+0.5ASR* as well as *0.9MT+0.1ASR* in the results). It is important to note that WCE systems are not retrained here since we perform a late fusion of scores from two different systems. Training a specific WCE system for SLT based on joint ASR and MT features is part of future work.

The results of these 3 systems are given in the last 3 columns of Table 5. They are obtained on the whole test set [3]. For the late fusion (MT+ASR), we do an arithmetic mean of both WCE systems scores [4]. From these results, we see that the use of both ASR-based and MT-based confidence scores improve the F-score for *"B"* label from 39.34% (MT only features) and 38% (ASR only features) to 43.96% (MT+ASR features), while giving similar F-score for *"G"* label. It is also interesting to notice that using ASR features lead to reasonable performance, almost equivalent to the MT features baseline. This can appear as rather disturbing because in that case, WCE estimator do not look at the translation to predict the confidence of the target words ; it only uses (detected) ASR errors to decide which word is good or bad in the speech translation output.

Figure 1 reports more detailed experiments where the G/B decision threshold varies systematically from 0.5 to 0.9 (with a step of 0.025). The different systems use different linear combination weights.

- *Weight=1* corresponds to the use of MT features only,
- *Weight=0.9* linearly combines both confidence scores as follows: 0.9MT+0.1ASR (intuitively, we thought that MT features would be more important),
- *Weight=0.5* linearly combines both confidence scores as follows: 0.5MT+0.5ASR,
- *Weight=0* corresponds to the use of ASR features only,

[3] They are given to illustrate how our database can be used, with basic strategies to fuse ASR and MT scores. More advanced fusion together with a crossvalidation protocol will be presented in future work

[4] All the results of the table are given using a G/B decision thershold which is a priori set to 0.7

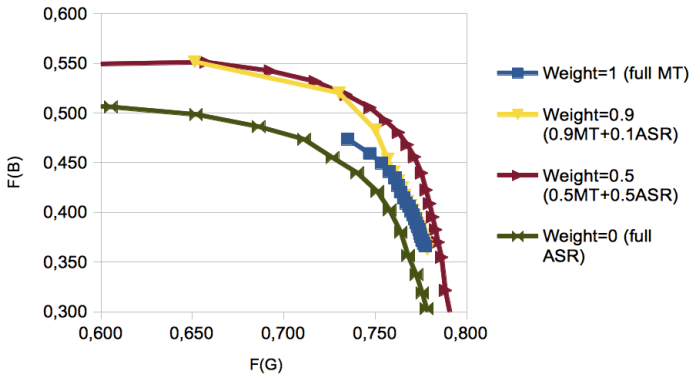


Figure 1: WCE performance (F(B) vs F(G) of different WCE methods - for SLT - for different decision thresholds varying from 0.5 to 0.9).

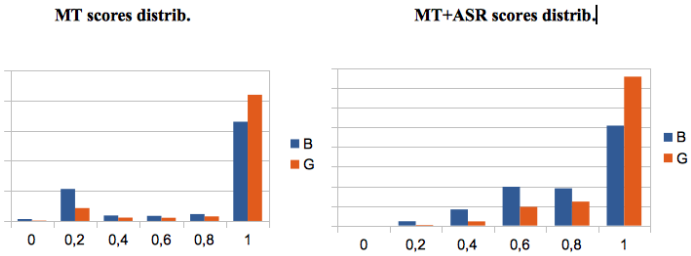


Figure 2: Evolution of the WCE scores distribution from MT features to MT+ASR features

From this figure, we see clearly that using both MT and ASR confidence scores improves the overall WCE performance. However, looking at the results obtained separately by the individual systems, one would have expected a better improvement with their combination. One explanation for this is the fact our WCE scores distributions are rather biased (as seen in Figure 2, many scores equal 1 for both G and B labels). Even if averaging (or linearly combining) ASR and MT scores tend to improve the class separability (Figure 2 shows how the WCE scores distributions evolve from *MT* to *MT+ASR* features), a better strategy might be to replace linear combination by more advanced strategies such as decision trees, SVMs or joint classifier based on the union of ASR and MT features, etc.

6. Conclusion

We presented a specific corpus to study and evaluate word confidence estimation of speech translation. It contains 2643 speech utterances with a quintuplet containing ASR output, verbatim transcript, MT output, SLT output and post-edition of translations. Researchers interested in making use of the dataset can download it from *github.com/besacier/WCE-SLT-LIG*. We also intend to record speech for the 10000 sentences of the train part described in Table 2. The perspectives of this work are numerous:

- propose a new shared task on word confiedence estimation for speech translation,
- train a single WCE system for SLT using joint ASR+MT features and see if more SLT errors can be accurately detected,
- rescore speech translation N-best lists or redecode speech translation graphs using WCE information, as was done by [2] but for MT only,
- use WCE for data augmentation from un-transcribed (and/or un-translated) speech in semi-supervised SLT scenarios,
- adapt WCE system for real interactive speech translation scenarios such as news or lectures subtitling,
- move from a binary (Good or Bad translation) to a 3-class decision problem (Good, ASR error, MT error),
- study how WCE can be adapted to a simultaneous interpetation task.

7. References

[1] N.-Q. Luong, L. Besacier, and B. Lecouteux, "Word Confidence Estimation for SMT N-best List Reranking," in *Proceedings of the Workshop on Humans and Computer-assisted Translation (HaCaT) during EACL*, Gothenburg, Suède, 2014. [Online]. Available: http://hal.inria.fr/hal-00953719

[2] N. Q. Luong, L. Besacier, and B. Lecouteux, "An Efficient Two-Pass Decoder for SMT Using Word Confidence Estimation," in *European Association for Machine Translation (EAMT)*, Dubrovnik, Croatie, jun 2014. [Online]. Available: http://hal.inria.fr/hal-01002922

[3] M. Snover, N. Madnani, B. Dorr, and R. Schwartz, "Terp system description," in *MetricsMATR workshop at AMTA*, 2008.

[4] M. Potet, L. Besacier, and H. Blanchon, "The lig machine translation system for wmt 2010," in *Proceedings of the joint fifth Workshop on Statistical Machine Translation and Metrics MATR (WMT2010)*, A. Workshop, Ed., Uppsala, Sweden, 11-17 July 2010.

[5] M. Potet, R. Emmanuelle E, L. Besacier, and H. Blanchon, "Collection of a large database of french-english smt output corrections," in *Proceedings of the eighth international conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey, May 2012.

[6] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[7] S. Galliano, E. Geoffrois, G. Gravier, J. F. Bonastre, D. Mostefa, and K. Choukri, "Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news," in *In Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC 2006*, 2006, pp. 315–320.

[8] A. Asadi, R. Schwartz, and J. Makhoul, "Automatic detection of new words in a large vocabulary continuous speech recognition system," *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 1990.

[9] S. R. Young, "Recognition confidence measures: Detection of misrecognitions and out-of-vocabulary words," *Proc. of International Conference on Acoustics, Speech and Signal Processing*, pp. 21–24, 1994.

[10] B. Lecouteux, G. Linarès, and B. Favre, "Combined low level and high level features for out-of-vocabulary word detection," *INTERSPEECH*, 2009.

[11] T. Kemp and T. Schaaf, "Estimating confidence using word lattices," *Proc. of European Conference on Speech Communication Technology*, pp. 827–830, 1997.

[12] J. Fayolle, F. Moreau, C. Raymond, G. Gravier, and P. Gros, "Crf-based combination of contextual features to improve a posteriori word-level confidence measures." in *Interspeech*, 2010.

[13] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Communication*, 2004.

[14] C. R. Antoine Laurent, Nathalie Camelin, "Boosting bonsai trees for efficient features combination : application to speaker role identification," in *Interspeech*, 2014.

[15] L. F. Lamel, J.-L. Gauvain, M. Eskénazi, *et al.*, "Bref, a large vocabulary spoken corpus for french1," *training*, vol. 22, no. 28, p. 50, 1991.

[16] A. L.-F. Han, Y. Lu, D. F. Wong, L. S. Chao, L. He, and J. Xing, "Quality estimation for machine translation using the joint method of evaluation criteria and statistical modeling," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 365–372. [Online]. Available: http://www.aclweb.org/anthology/W13-2245

[17] N. Q. Luong, B. Lecouteux, and L. Besacier, "LIG system for WMT13 QE task: Investigating the usefulness of features in word confidence estimation for MT," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 396–391.

[18] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting et labeling sequence data," in *Proceedings of ICML-01*, 2001, pp. 282–289.

[19] E. Bicici, "Referential translation machines for quality estimation," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 343–351. [Online]. Available: http://www.aclweb.org/anthology/W13-2242

[20] T. Lavergne, O. Cappé, and F. Yvon, "Practical very large scale crfs," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 504–513.

[21] N. Bach, F. Huang, and Y. Al-Onaizan, "Goodness: A method for measuring machine translation confidence," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, June 19-24 2011, pp. 211–219.

[22] N. Ueffing, K. Macherey, and H. Ney, "Confidence measures for statistical machine translation," in *Proceedings of the MT Summit IX*, New Orleans, LA, September 2003, pp. 394–401.

[23] N. Q. Luong, L. Besacier, and B. Lecouteux, "Word confidence estimation and its integration in sentence quality estimation for machine translation," in *Proceedings of The Fifth International Conference on Knowledge and Systems Engineering (KSE 2013)*, Hanoi, Vietnam, October 17-19 2013.

[24] D. Xiong, M. Zhang, and H. Li, "Error detection for statistical machine translation using linguistic features," in *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*, Uppsala, Sweden, July 2010, pp. 604–611.

Machine Translation of Multi-party Meetings: Segmentation and Disfluency Removal Strategies

Eunah Cho, Jan Niehues, Alex Waibel

International Center for Advanced Communication Technologies - InterACT
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology, Germany
{eunah.cho|jan.niehues|alex.waibel}@kit.edu

Abstract

Translating meetings presents a challenge since multi-speaker speech shows a variety of disfluencies. In this paper we investigate the importance of transforming speech into well-written input prior to translating multi-party meetings. We first analyze the characteristics of this data and establish oracle scores. Sentence segmentation and punctuation are performed using a language model, turn information, or a monolingual translation system. Disfluencies are removed by a CRF model trained on in-domain and out-of-domain data. For comparison, we build a combined CRF model for punctuation insertion and disfluency removal. By applying these models, multi-party meetings are transformed into fluent input for machine translation. We evaluate the models with regard to translation performance and are able to achieve an improvement of 2.1 to 4.9 BLEU points depending on the availability of turn information.

1. Introduction

Machine translation (MT) of spontaneous speech has recently drawn a great deal of interest. For instance, the importance of sentence segmentation, punctuation insertion and disfluency removal for translating monologue data, such as lectures, has been researched extensively. In addition, there have been research efforts investigating MT of two-party speech, such as telephone calls. However, automatic translation of multispeaker speech remains yet underexplored.

In our globalized world, teams of different parts of the world are increasingly working together. Internal team meetings held in one language need to be translated into another language in order to make the discussions available to all involved parties. Human translation is time-consuming and costly, so MT can be a supportive tool to overcome this challenge. State-of-the-art MT systems, however, are not designed for such conversational speech, especially when multiple speakers are involved. Since conventional MT systems are built using written texts, their performance drops when they are applied to such a different domain. We therefore propose an approach to transform multi-party meetings so they are closer in style to the training data of the MT system.

Natural language processing (NLP) of multispeaker speech presents unique research challenges. Speech disfluencies should be removed, while the punctuation marks and sentence boundaries need to be inserted.

Spontaneous speech contains a large number of disfluencies, such as hesitations as well as repetitions, either exactly or vaguely the same, and speech fragments. In addition to these disfluencies, speakers may interrupt each other. Due to such interruptions, aborted speech fragments occur very often in multispeaker speech. Therefore, it is one of our main goals to model such disfluencies which can better fit the domain. One of the difficulties of disfluency detection, however, is data sparsity, since speech disfluencies are usually modelled using disfluency-annotated data. Thus it is necessary to explore how to improve the performance given the limited quantity of data as well as evaluate how important the domain is for the given task.

Since the output of an automatic speech recognition (ASR) system is a stream of word tokens, without punctuation or segmentation information, it is necessary to properly segment and punctuate the ASR output for translation.

In this work, we present various approaches to reformulate multispeaker speech prior to MT, through segmentation, punctuation insertion and disfluency removal. In order to explore the importance of domain in this task, we train disfluency removal models on in-domain and out-of-domain data and compare the results. Every experiment is conducted in two conditions whether turn information is available or not. Once the disfluencies of the meeting data are removed and punctuation marks are inserted, the data goes through our English to French MT system. For comparison, oracle experiments results and a baseline system are shown.

2. Related Work

There has been extensive effort on disfluency removal on telephone speech, or Switchboard data [1]. In [2], Johnson et al. combined the noisy channel approach with a tree adjoining grammar for modeling speech disfluencies. In the noisy channel model, it is assumed that fluent text goes through a channel which adds disfluencies. Disfluency removal on

the same data is modeled using a conditional random fields (CRF) model in [3], using language and lexical model, and parser information as features.

In [4], segmentation and disfluency removal issue in meeting data is handled in the scope of ASR. Baron et al. explored sentence boundary and disfluency detection in meetings using prosodic and lexical cues. For multi-party meeting data they used data collected as part of the ICSI Meeting Recording Project [5]. Sentence boundary detection is treated as a sequence classification problem, where each word boundary is labeled as either a sentence boundary, a disfluency interruption point, or a clean word transition. Therefore, disfluency is viewed from a different perspective, as an interruption point, where once it occurs a new segment boundary is added. Baron et al. find that combining prosodic and word-based classifier information yields the best results for the given task.

While the previous works have focused on enhancing the performance of speech recognition, Peitz et al. [6] compared the translation performance using three different methods to punctuate TED talks. They compare methods depending on when and how the punctuation marks are inserted: prediction in the source language, implicit prediction, and prediction in the target language. They assumed that the proper segments are already available, but punctuation marks are missing therefore should be inserted. Among the three systems, translating from unpunctuated to punctuated text achieves the largest improvements. Later this work is extended in [7] for MT of university lectures, where a monolingual translation system is used for punctuation combined with sentence boundary detection. They prepare the training data by cutting it randomly, so that detection of sentence-like units is possible.

Cho et al. [8] use a monolingual translation system together with CRF-based disfluency removal. Using a CRF model, the disfluency probability of each token is obtained and encoded into word lattices so that potentially disfluent paths can be skipped during decoding.

MT of multi-party meetings was studied in [9], with a particular view towards analyzing the importance of modeling contextual factors. They showed that word sense disambiguation using topic and domain knowledge yields a large improvement on MT performance.

Recently Hassan et al. [10] investigated the impact of segmentation and disfluency removal on translation of telephone speech. They use a CRF model to detect sentence units and a knowledge-based parser for complex disfluency removal.

There are several notable differences between our and previous work. Contrary to many works in disfluency removal and punctuation insertion, our work is expanded to the MT. Our systems are designed for multi-party meetings unlike [7, 10]. We focused on segmentation and disfluency issues in multi-party meetings, while [9] studied the meetings with focus on word sense disambiguation. Additionally, the importance of training the models on out-of-domain data is investigated in our work.

3. Task

Before describing the techniques to translate multispeaker speech, the corpus and its characteristics are described. The section is concluded with an overview of the system architecture to detect speech disfluencies and punctuation marks used in this evaluation.

3.1. Multi-party meeting data

Our corpus consists of project meetings between project participants with various topics. We use eight sessions, where each meeting involves 5 to 12 different speakers. All meetings are held in English. As in real meeting scenarios, the meeting participants consist of native and non-native English speakers. The eight meeting sessions are transcribed and then disfluencies are manually annotated. We use five of the meetings for training the disfluency removal model and the remaining three for testing. The test data is translated into French in order to evaluate the translation performance.

3.1.1. Speech disfluencies

Disfluencies in the meeting data are annotated manually by human annotators. Previous work on disfluencies [2, 11, 12] categorized the disfluencies into three groups: `filler`, `(rough)copy`, and `non-copy`. `filler` contains filler words as well as discourse markers. Therefore, this class includes words such as *uh*, *you know*, and *well* in some cases. As the class name suggests, `(rough)copy` includes an exact or rough repetition of words or phrases. In spontaneous speech, speakers may repeat what has been already spoken, as stutter or correction. For example, a sentence *There is, there was an advantage* has `(rough)copy` in the phrase *there is*. `non-copy` includes the cases where the speaker aborts previously spoken segments and starts a new segment. It can be rather moderate, so that the newly started fragment still has the same theme as the previously spoken segment. In a more extreme case, however, the speaker may introduce an entirely different topic in the new fragment. For example, in the following sentence from our meeting data: *I don't think it's the, the crucial thing is that we can compile with...*, the part before the comma is annotated as `non-copy`.

After looking into the data, we decided that the disfluency annotations for the multispeaker speech task has an additional category, `interruption`. While the other three categories of disfluency can be used for other tasks such as monologue, the last class `interruption` is devised for this new task. In multispeaker speech, generally there are more than two speakers involved. Therefore, there are many parts of utterances which are interrupted by other speakers. Those segments which are interrupted and therefore could not be finished were classified as `interruption`.

The number of tokens of each class of disfluencies and its

Table 1: *Meeting data statistics*

Class	Training		Testing	
`filler`	2,666	6.9%	999	6.7%
`(rough)copy`	2,232	5.8%	1,017	6.8%
`non-copy`	802	2.1%	331	2.2%
`interruption`	1,350	3.5%	864	5.8%
`clean`	31,507	81.7%	11,660	78.4%
SUM	38,557	100%	14,871	100%

proportion are shown in Table 1. The numbers do not include punctuation marks, but only words. Both the training and test data have around a disfluency rate of around 20%, which is much higher than the rate reported in [13], where lecture data has a disfluency rate of roughly 10%. Around 7% of the word tokens in the meeting data are simple disfluencies, or filler words, while the other 11 to 15% are more difficult to detect.

3.1.2. Segments

The training data shown in Table 1 consists of $4.6k$ sentences, while the test data has around $2.1k$ sentences. We found that multi-party meeting data has the characteristic that each segment is rather short. In average, for all meeting data we have, there are around 8 words per segment. This is quite short compared to, for example, lecture data, which has around 15 words per segment [13]. We also compare the number of segments to the training data of our MT system, which is mainly parliamentary proceedings and news text. This data has around 24 words per segment.

Figure 1: *Statistics on number of words in segment*

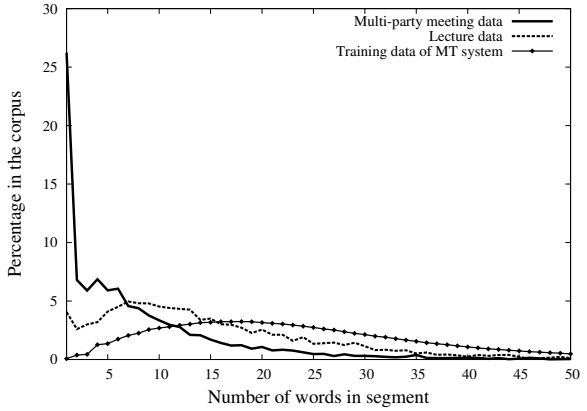


Figure 1 depicts the distribution of segment length for every corpus. In the meeting data short segments are the majority, especially one word segments. There are many segments which only consist of one word, such as *yes* or *okay*. Although some of them are discourse markers and therefore annotated as `filler` disfluency, some of them are also left intact when those tokens are actually used to convey meaning. This is therefore another challenge of detecting disfluencies in meeting data. Another cause of the short segments is that there are also many short segments which are interrupted by another speaker and therefore aborted. The lecture data, which also consists of spoken language, also has higher frequency of shorter segments, compared to the conventional MT training data which has more segments whose length is longer than 15 words.

3.1.3. Example

Table 2 shows an example excerpt from the meeting data. Following the annotation conventions described in [13], `filler` tokens are marked with $<>$, and `(rough)copy` tokens are marked with $+//+$. `non-copy` tokens are tagged with $-//-$, and finally `interruption` are marked with $\#//\#$. In this excerpt, the first speaker tried to start a new fragment (starting *what*), then a filler word is occurred (*uh*), and then the fragment is aborted, then yet another fragment is started (*how far*). But this last fragment is interrupted by the next speaker. We can also observe repetition.

Table 2: *Meeting data example with disfluency annotation*

A: I haven't heard anything, so I don't know -/what/-
<uh> #/how far/#
B: I will check for that.
C: Why is the API so hard?
We're waiting for a month now for this.
D: I don't know +/the last/+ the last meeting outcome <uh>
he said he could give us API at the end of the month.
C: Okay.

3.2. System architecture

In this work, we chose a work scheme where the output stream from an ASR system passes first through an automatic disfluency detection system. Based on this cleaned-up stream, punctuation and segmentation insertion is performed. Once the disfluencies in the ASR output are removed and punctuation marks are inserted, the cleaned, punctuated data goes through the MT system like normal input data.

Disfluency detection is performed prior to the punctuation and segmentation insertion, because this way punctuation insertion can be trained on much larger data. While disfluency removal can be only trained on disfluency-annotated data, punctuation insertion can be trained on more data. For the disfluency removal model, we use data of two different domains: multi-party meeting and lecture. As the first domain, we train the model using five meeting sessions, which sum up to $38.6k$ annotated words. In order to model the case where we have no in-domain data, we train the second model using lecture data. We use web-based seminar lecture data

given in English as well as the annotated English reference translation of the German lecture data shown in [13]. The lecture data sums up to $104k$ annotated words, and shows a moderate level of disfluency.

The punctuation insertion model is not trained using the meeting data, but using the English side of the MT training corpus, which consists of well-segmented, clean text.

Once the models are built, they are applied to the remaining three meeting sessions. The test data consists of $2.1k$ segments with $14.9k$ English words and $11.4k$ French words. After cleaning up the disfluencies manually, the source side contains $11.7k$ English words.

3.2.1. Turn information

For MT of multi-party meetings, turn information can play a big role, since knowing who spoke when can provide basic segmentation. However, turn information is not always available.

In order to compare and study the impact of turn information on our models, we assume two scenarios: in the first scenario turn information is available while in the second one it is not available. With the turn information, basic segment information according to speaker changes is available. Even though this may not be the exact sentence segmentation, it can offer a reasonable baseline for segmentation and punctuation insertion. It can also offer additional features for disfluency detection. As it is possible to know which segment is started by which speaker, we can obtain a cue that the previous segments' last tokens could have been interrupted by the new speaker, given the fact that meetings contain a lot of interruptions.

When the turn information is not available, there is no basic segmentation. Therefore it is required to chunk the stream of ASR output into segments. Different tactics on segmentation and punctuation insertion will be described in Section 5.

4. Disfluency Detection

In the disfluency detection model, we start with a sequence of words as input and need to mark parts of the sequence as disfluencies. This problem can intuitively be modeled as a sequence labeling task, where each word is either labeled by one of the disfluency classes (`filler`, `(rough)copy`, `non-copy`, and `interruption`), or by a label representing clean speech. Since sequence labeling is a common problem in NLP, it has been studied intensively. One succesful approach to model these problems is using CRF. As CRFs can represent long-range dependencies in the observations, they have shown good performance in sentence segmentation [14], parts of speech (POS) tagging [15] and shallow parsing [16]. In this work we use the CRF model implemented in the GRMM package [17] to mark the speech disfluencies. The CRF model was trained using L-BFGS, with the default parameters of the toolkit.

4.1. In-domain vs. out-of-domain data

In the ideal case, disfluency annotated in-domain data is available for training the CRF model. However, the annotation of speech for different domains can be very time-consuming. As disfluency annotated lecture data [13] is available, we use this data as our out-of-domain training data for the CRF model. As in-domain training data we use the in-house English meeting data. This will show whether the disfluency removal model is portable across different domains.

Compared to the meeting data, lecture data has different characteristics. Although it still provides general speech disfluencies such as repetitions or filler words, lecture data in general contains a moderate level of speech disfluencies compared to the quite noisy meeting data. Especially, unlike meeting data, lecture data does not contain interruptions by other speakers. Therefore, for testing the CRF model using lecture data, we mapped `interruption` onto the `non-copy` class.

As a test data of the CRF model, we use the test data described in Section 3. After potential disfluencies are detected and removed, punctuation and segmentation are inserted into this test set, which is then used as input for MT.

4.2. Features

As features for CRF, we use lexical and language model (LM) features inspired by the work in [11]. Lexical features include current and adjacent words/POS tokens, whether the current word is a partial word, and whether words or POS tokens are showing repetitive patterns. LM features include unigram and 4-gram LM scores, and their ratio. In addition to these features, following [12], features obtained from a word representation in vectors and phrase table information are used. Each word is represented as a word vector with 100 dimensions as shown in [18]. Afterwards the vectors are clustered into 100 clusters using the k-means algorithm. We use the cluster number of each word as one of the features, as well as the repetitive pattern of the cluster code and adjacent words' cluster codes. For the phrase table information, we use the phrase table which is used for the actual MT of the task and check the potential translations of each word.

As mentioned earlier, we assumed two scenarios about turn information availability. In the scenario where the turn information is available, we extracted the word position within the turn. We expect that disfluencies can be more prominent in the initial part of each turn, because many stutters as well as corrections occur within the first several words. In addition, as interruptions between speakers occur at end of each turn, we encoded whether the current token is one of the first or final 5 words of the turn in order to incorporate this information for the training.

The CRF model is trained with a bigram feature, so that first-order dependencies between words with a disfluency can be modeled.

5. Segmentation and Punctuation Insertion

After removing disfluencies, the main difference between written text and the disfluency-removed speech is the lack of punctuation marks. In recent work [6], it has been shown that a promising approach to translate unpunctuated text is to automatically insert punctuation marks and segmentation prior to translation. Therefore, we analyzed three different methods to segment and punctuate the multi-party meeting data: simple LM-based segmentation, turn segmentation, and monolingual translation system.

5.1. Simple LM-based segmentation

Assuming there is no information about different speakers and their turns available, ASR of such a talk would generate a stream of words. For translation, it is necessary to segment the stream of words. As a baseline system, we segmented based on a hard threshold of word-based LM scores. First we concatenated the test data into a single line without any punctuation marks, in order to mimic the ASR output. We use a 4-gram LM trained on the punctuated English side of the MT training corpus and measure the probability of a final period given the previous words. When the probability exceeded an empirically chosen threshold, we inserted a final period and started a new segment. The output of this baseline system consists of segments where each segment ends with a final period.

5.2. Turn segmentation

If we have access to turn information, we can exploit this information in order to obtain a better baseline segmentation. We inserted a final period and began a new segment whenever the speaker changed. Each segment of this system may contain more than one actual sentence, with no further punctuation marks within the segment.

5.3. Monolingual translation system

Cho et al. [7] successfully used a monolingual translation system to insert punctuation marks into non-punctuated German lecture data. Following this approach, we built a monolingual translation system from non-punctuated English to punctuated English. While the previous two methods insert only final periods, this system can insert all punctuation marks appeared in the training data. As training data we used the English side of the MT training corpus. This MT training corpus is ideally segmented and contains all punctuation marks, including a final period at the end of each sentence. In order to learn where segment breaks should be inserted, we throw away the segmentation and randomly cut the English side of the data. Aiming to generate data that is similar to the test data, we limit the length of segments to 22 words. The test data goes through the monolingual translation system with a sliding window of 10 words.

For the scenario where turn information is available, we build an additional, slightly different monolingual translation system. When we have the turn information, several segments uttered by a speaker are concatenated. Therefore, in order to make the training data similar to the test data, we concatenated one to three sentences randomly into one sentence. Punctuation marks between sentences are removed, and only a final period is added at the end of each line of the source side data. The target side contains all punctuation marks.

6. Experiments

In this section, we briefly describe the MT system we use in our experiments. Oracle experiments and the results are given, followed by results of segmentation and punctuation insertion. The results of disfluency removal are analyzed. Finally, the overview of our system is given in the end.

6.1. System description

The translation system is trained on 2.3 million sentences of English-French parallel data including the European Parliament data and the News Commentary corpus. The parallel TED data[1] is used as in-domain data for the MT models. As development data, we use manual transcripts of TED data.

Preprocessing which consists of text normalization and tokenization is applied before the training. In order to build the phrase table, we use the Moses package [19]. Using the SRILM Toolkit [20], a 4-gram language model is trained on 683 million words from the French side of the data. A bilingual language model [21] is used to extend source word context. The POS-based reordering model as described in [22] is applied to address different word orders between English and French. We use Minimum Error Rate Training (MERT) [23] for the optimization in the phrase-based decoder [24]. All scores of translation into French are reported in case-sensitive BLEU scores [25] in this paper. When the sentence boundaries differ from the reference translation, we use the Levenshtein minimum edit distance algorithm [26] to align hypothesis for evaluation.

6.2. Oracle experiments

Table 3 shows the translation performance for oracle punctuation marks and oracle disfluency removal on the multi-party meeting data.

Table 3: *Oracle experiments*

System	No turns	Turns
Baseline	9.53	12.93
Oracle segmentation	13.96	
Oracle punctuation	15.64	
Oracle disfluency	12.21	15.72
Oracle all	20.93	

[1] `http://www.ted.com`

In the first system, all disfluencies are kept and baseline segmentations are used. As the baseline segments, we use two different segmentation methods. When there is no turn information available, segmentation and final periods are inserted using the simple LM-based method as described in Section 5.1. On the other hand, when we have access to the turn information, a new segment and a final period are inserted whenever the speaker changes as described in Section 5.2. We can observe that using the turn information is very helpful in achieving better performance.

Then we insert oracle segmentation and a final period at the end of segment. When we also inserted all other punctuation marks from the reference transcript, the translation performance is improved up to 15.64 BLEU points even though it still contains all disfluencies. We can observe that nearly 1.7 BLEU points are achieved by inserting all other punctuation marks, on top of we have the ideal reference segmentation and a final period.

In the next experiment, we keep the punctuation and segmentation the same as in the baseline system, but remove all of the manually annotated disfluencies. By doing so, translation performance is improved by around 3 BLEU points compared to the baseline system. Finally, we achieved BLEU score of 20.93 when we have the oracle for both punctuation and disfluency. This is the upper bound of the performance we can get for this test set when we have both perfect segmentation/punctuation and disfluency removal.

As shown by these numbers, the performance can be improved by more than 10 BLEU points if the ideal punctuation and disfluency detection are applied. Therefore, modeling these two problems in a translation system of mutispeaker speech is essential to reach a good translation quality.

6.3. Segmentation and punctuation insertion

In this section, we look into the performance of the segmentation and punctuation in a realistic approach (all disfluencies kept) and perfect conditions (remove all disfluencies using the manual annotation).

Table 4: *Punctuation insertion, no turn information*

System	Keep disf.	Oracle disf.
Baseline	9.53	12.21
Mono. trans.	12.44	16.34
Oracle punctuation	15.64	20.93

Table 4 shows the results under the assumption that no turn information is available. The baseline system has punctuation and segmentation inserted using the simple LM-based method. When punctuation marks are inserted using the monolingual translation system, we achieved an improvement of 3 to 4 BLEU points for both disfluency conditions. This improvement reaches almost half of the difference between the baseline systems and oracle scores. We can also observe that when segmentation and punctuation are improved, the impact of disfluencies increases. There is bigger room of improvement which can be achieved by removing correct disfluencies, when we have better segmentation and punctuation. The same phenomena can be observed in the experiments with turn information, as shown in Table 5.

Table 5: *Punctuation insertion, with turn information*

System	Keep disf.	Oracle disf.
Baseline	12.93	15.72
Mono. trans.	13.25	17.71
Oracle punctuation	15.64	20.93

We can observe that the baseline scores in this case have already improved a lot over the experiments without turn information. Since the baseline segmentation is already better, the improvements are smaller, but there are still consistent improvements when inserting punctuation marks using the monolingual translation system.

6.4. Disfluency removal

This section presents translation performance when we apply the disfluency removal models trained either on in-domain or out-of-domain data. Punctuation and segmentation are inserted not only by the monolingual translation system for the realistic case, but also oracle punctuation is used for comparison.

Table 6: *Disfluency removal, no turn information*

System	Mono. trans.	Oracle punct.
Keep disfluency	12.44	15.64
CRF in-domain	14.41	17.26
CRF out-of-domain	14.24	16.95
Oracle disfluency	16.34	20.93

Table 6 shows the scores under the assumption that there is no turn information available. In the first experiment, we keep all disfluencies. Then we show the scores when we use the disfluency removal model trained only on the in-domain data, multi-party meeting data. These scores are compared with the scores when we use the model trained only on the out-of-domain data, which is lecture data. Finally, we show the scores removing all disfluencies annotated. An interesting point is that using lecture data for training the CRF model yields similar performance to training using the meeting data. Even though using the lecture data is slightly worse than using the meeting data, the difference is minimal.

Our preliminary experiments showed that when we use the in-domain data for training the disfluency removal model, we have around 8 points better F-scores, compared to the case when we train the model using out-of-domain data. However, such differences are not pronounced in terms of BLEU. It shows that the disfluency modeling technique

shown in this work can be transfered into a new domain without causing a big loss of performance in MT.

Table 7: *Disfluency removal, with turn information*

System	Mono. trans.	Oracle punct.
Keep disfluency	13.25	15.64
CRF in-domain	15.01	17.10
CRF out-of-domain	14.90	17.03
Oracle disfluency	16.34	20.93

This result is also observable when the models are trained with turn information, as shown in Table 7. The disfluency removal model trained on meeting data performs only slightly better than the lecture data. In all listed conditions, it is shown that we can improve the translation quality by 1.5 to 2 BLEU points by removing disfluencies.

6.5. Combined modeling of punctuation insertion and disfluency removal

As an additional experiment, we model punctuation marks and disfluencies in one model. This yields the advantage that it is not necessary for ASR output to pass through two different steps. We also hope that this experiment can provide the first insight on MT performance when modeling these two in one model for the given task. In this scheme, both the punctuation marks as well as disfluencies are predicted given the potentially disfluency, and unpunctuated ASR output. For modeling we use the same features as for the disfluency removal. Thus, punctuation and disfluencies are trained using the data with speech disfluencies. For the modeling, we use the same CRF tool, but with two decision labels: one with disfluency classes and another one with punctuation marks.

Table 8: *Punctuation insertion and disfluency removal in one model*

System	No turn	Turn
Baseline	9.53	12.93
Combined CRF in-domain	13.92	14.45
CRF in-domain + Mono. trans.	14.41	15.01
Combined CRF out-of-domain	13.99	14.58
CRF out-of-domain + Mono. trans.	14.24	14.90
Oracle all	20.93	

Table 8 presents the results of this experiment. When modeling punctuation marks and disfluency removal together in one model, it still provides a big improvement over the baseline, where all disfluencies are kept. Same as in the previous experiments, training the models on in-domain or out-of-domain data does not cause a big performance difference in MT. Comparing the scores of training the models separately for disfluencies and punctuation marks, however, the scores are generally around 0.3 to 0.5 BLEU points worse. The F-score of disfluency removal does not get affected significantly even when we are modeling it along with punctuation marks. However, as the monolingual translation system is trained using much more data, the performance of segmentation and punctuation insertion is affected and therefore degrades the overall performance.

6.6. Overview

Finally, Table 9 shows the best scores achieved in this work.

Table 9: *Overview*

System	No turn	Turn
Baseline	9.53	12.93
Best system	14.41	15.01
Oracle	20.93	

In our best system we first remove disfluencies using a CRF model trained on the in-domain data, and then insert proper segmentation and punctuation marks using the monolingual translation system. When there is no turn information, we achieve around 4.9 BLEU points of improvement. With turn information, we improve the system by around 2.1 BLEU points.

7. Conclusion

In this paper, we showed how machine translation performance is affected when different techniques for segmentation, punctuation insertion and disfluency removal are applied to multispeaker speech. The characteristics and differences of multispeaker speech compared to other data were described. We built two separate disfluency removal systems using in-domain and out-of-domain data and their performances are compared in terms of translation quality. We showed that our disfluency removal technique presented in this work can be transfered to a new domain. Segmentation and punctuation insertion systems are applied after the disfluencies are removed. The best system of disfluency removal and punctuation detection models achieves a gain of 4.9 BLEU points when there is no turn information and 2.1 BLEU points when turn information is available over the baseline. As an additional experiment, a sequence tagging model which models both segmentation, punctuation insertion and disfluency removal is built and the performance is compared to our best automatic systems.

In future work, we would like to explore integrating segmentation, punctuation insertion and disfluency removal systems into end-to-end speech translation systems for real-time evaluation.

8. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

9. References

[1] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone Speech Corpus for Research and Development," in *ICASSP*, San Francisco, CA, USA, 1992.

[2] M. Johnson and E. Charniak, "A TAG-based Noisy Channel Model of Speech Repairs," in *ACL*, Barcelona, Spain, 2004.

[3] E. Fitzgerald, F. Jelinek, and R. Frank, "What Lies Beneath: Semantic and Syntactic Analysis of Manually Reconstructed Spontaneous Speech," in *ACL*, Singapore, 2009.

[4] D. Baron, E. Shriberg, and A. Stolcke, "Automatic Punctuation and Disfluency Detection in Multi-party Meetings using Prosodic and Lexical Cues," in *ICSLP*, Denver, CO, USA, 2002.

[5] N. Moran, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke, "The Meeting Project at ICSI," in *HLT*, San Diego, CA, USA, 2001.

[6] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling Punctuation Prediction as Machine Translation," in *IWSLT*, San Francisco, CA, USA, 2011.

[7] E. Cho, J. Niehues, and A. Waibel, "Segmentation and Punctuation Prediction in Speech Language Translation using a Monolingual Translation System," in *IWSLT*, Hong Kong, China, 2012.

[8] ——, "Tight Integration of Speech Disfluency Removal into SMT," in *EACL*, Gothenburg, Sweden, 2014.

[9] Y. Mei and K. Kirchhoff, "Contextual Modeling for Meeting Translation Using Unsupervised Word Sense Disambiguation," in *Coling*, Beijing, China, 2010.

[10] H. Hassan, L. Schwartz, D. Hakkani-Tür, and G. Tur, "Segmentation and Disfluency Removal for Conversational Speech Translation," in *Interspeech*, September 2014.

[11] E. Fitzgerald, K. Hall, and F. Jelinek, "Reconstructing False Start Errors in Spontaneous Speech Text," in *EACL*, Athens, Greece, 2009.

[12] E. Cho, T.-L. Ha, and A. Waibel, "CRF-based Disfluency Detection using Semantic Features for German to English Spoken Language Translation," in *IWSLT*, Heidelberg, Germany, 2013.

[13] E. Cho, S. Fünfer, S. Stüker, and A. Waibel, "A Corpus of Spontaneous Speech in Lectures: The KIT Lecture Corpus for Spoken Language Processing and Translation," in *LREC*, Reykjavik, Iceland, 2014.

[14] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using Conditional Random Fields for Sentence Boundary Detection in Speech," in *ACL*, Ann Arbor, MI, USA, 2005.

[15] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilitic Models for Segmenting and Labeling Sequence Data," in *ICML*, Williamstown, MA, USA, 2001.

[16] F. Sha and F. Pereira, "Shallow Parsing with Conditional Random Fields," in *HLT/NAACL*, Edmonton, Canada, 2003.

[17] C. Sutton, "GRMM: A Graphical Models Toolkit," 2006. [Online]. Available: http://mallet.cs.umass.edu

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Workshop at ICLR*, Scottsdale, AZ, USA, 2013.

[19] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *ACL, Demonstration Session*, Prague, Czech Republic, 2007.

[20] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *ICSLP*, Denver, CO, USA, 2002.

[21] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *WMT*, Edinburgh, UK, 2011.

[22] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *TMI*, Skövde, Sweden, 2007.

[23] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *WPT*, Ann Arbor, MI, USA, 2005.

[24] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *Int. Conf. on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation." IBM Research Division, T. J. Watson Research Center, Tech. Rep. RC22176 (W0109-022), 2002.

[26] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating Machine Translation Output with Automatic Sentence Segmentation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Boulder, Colorado, USA, October 2005.

Empirical Dependency-Based Head Finalization for Statistical Chinese-, English-, and French-to-Myanmar (Burmese) Machine Translation

Chenchen Ding[†], *Ye Kyaw Thu*[‡], *Masao Utiyama*[‡], *Andrew Finch*[‡], *Eiichiro Sumita*[‡]

[†]Department of Computer Science, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan
[‡]Multilingual Translation Laboratory,
National Institute of Information and Communications Technology
3-5 Hikaridai, Seikacho, Sorakugun, Kyoto, 619-0289, Japan
[†]tei@mibel.cs.tsukuba.ac.jp
[‡]{yekyawthu, mutiyama, andrew.finch, eiichiro.sumita}@nict.go.jp

Abstract

We conduct dependency-based head finalization for statistical machine translation (SMT) for Myanmar (Burmese). Although Myanmar is an understudied language, linguistically it is a head-final language with similar syntax to Japanese and Korean. So, applying the efficient techniques of Japanese and Korean processing to Myanmar is a natural idea. Our approach is a combination of two approaches. The first is a head-driven phrase structure grammar (HPSG) based head finalization for English-to-Japanese translation, the second is dependency-based pre-ordering originally designed for English-to-Korean translation. We experiment on Chinese-, English-, and French-to-Myanmar translation, using a statistical pre-ordering approach as a comparison method. Experimental results show the dependency-based head finalization was able to consistently improve a baseline SMT system, for different source languages and different segmentation schemes for the Myanmar language.

1. Introduction

The state-of-the-art techniques of statistical machine translation (SMT) [1, 2] demonstrate good performance on translation of languages with relatively similar word orders [3]. However, word reordering is a problematic issue for language pairs with significantly different word orders, such as the translation between a subject-verb-object (SVO) language and a subject-object-verb (SOV) language [4].

To resolve the word reordering problem in SMT, a line of research handles the word reordering as a separate preprocess, which is referred as *pre-ordering*. In pre-ordering, the word order on source-side is arranged into the target-side word order, before a standard SMT system is applied, on both training and decoding phases. The pre-ordering process can be realized in either a rule-based way or a statistical way. Generally, a rule-based approach needs a high-precision parser and effective manually designed rules; and a statistical approach needs data for model training.

An effective rule-based approach, *head finalization* has been proposed for English-to-Japanese translation [4]. The approach takes advantage of the *head final* property of Japanese on the target-side. It designs a head finalization rule to move the head word based on the parsing result by a head-driven phrase structure grammar (HPSG) parser. Generally, the idea can be applied to other SVO-to-Japanese translation tasks, such as its application in Chinese-to-Japanese translation [5]. However, an HPSG parser is not available for many languages, which prevents the HPSG-based head finalization from being applied to more languages. On the other hand, dependency parsers are available for more languages. A typical rule-based pre-ordering using dependency structure was proposed in [6]. Their approach used a rule set to arrange the order of a head word together with its modifiers.

In this paper, we explore dependency-based head finalization for an understudied language, Myanmar[1]. We use the dependency structure to realize the head finalization of [4]. Because the head finalization only moves a head word after all its modifiers, the proposed dependency-based head finalization is a simplified version of [6], which keeps the order of modifiers unchanged. So, our approach is simple and widely applicable for different source languages. On the target-side, there are no standard part-of-speech set and morpheme analysis tools available for Myanmar word segmentation yet, so we employ two word segmentation schemes: syllable-based and dictionary-based maximum matching. Experiments on Chinese-, English-, and French-to-Myanmar translation show that simple head finalization can efficiently and stably improve a baseline SMT system, no matter what the source-side language is or which segmentation scheme is used. We use a statistical pre-ordering approach [7] as a comparison method. We observe it performs well on certain situations, but it is sensitive to the source-side language and segmentation schemes.

[1]The language may be more referred as *Burmese* in English though, in this paper, we refer it consistently as *Myanmar*.

Figure 1: *Example of a Myanmar sentence “ သူသည် စာအုပ်ကို ဆရာအား ပေးသည် ” (English translation “he gives the book to the teacher”). The first row shows the morphemes in the Myanmar sentence, one-box-one-morpheme. Content morphemes are illustrated in black and functional morphemes are in gray. The second row is the English literal translations of them. In the two lower rows, Japanese and Korean translations of the Myanmar sentence are also shown, morpheme-by-morpheme. Both the Japanese and Korean sentences are grammatically correct, from which the syntactic similarity can be observed. The right-most boxes in Japanese and Korean sentences, which contain the verbs, should be noticed. The corresponding parts of Myanmar present marker in these two languages are inflection endings which cannot be detached from the verb stems (marked by gray, in the case of Korean, more correctly, the “ ㄴ다 ” part). While Myanmar has a completely detachable marker from the verb stem.*

This paper is organized as follows. In Section 2, we give an introduction to the Myanmar language. In Section 3, we discuss related approaches. In Section 4, we describe the proposed approach. In Section 5, we show the experimental results and present a discussion. Section 6 contains the conclusion and future work.

2. Myanmar Language

Myanmar is an SOV language that demonstrates a consistent head-final typology. Syntactically, Myanmar is quite similar to Japanese and Korean, where functional morphemes succeed content morphemes, and verb phrases succeed noun phrases. We show an example in Fig. 1 to show the features of Myanmar and its similarity to Japanese and Korean.

On the other hand, unlike Japanese and Korean, which are typical agglutinative languages, Myanmar is an analytic language, in which the morphemes are without inflection. This is because Myanmar is a monosyllabic language originally, where morphemes are only composed by non-inflected single syllables. Although Buddhism-related loanwords from the Pali language and modern loanwords from western languages have introduced polysyllabic morphemes into Myanmar, the basic framework of syntax has not been affected.

3. Related Work

As mentioned, Myanmar is an understudied language that has quite similar (or, even simpler) characteristics to Japanese and Korean, both of which are well studied. A natural idea is that we can transfer the Japanese or Korean language processing techniques to Myanmar.

HPSG-based head finalization [4] and dependency-based pre-ordering [6] are two typical rule-based pre-ordering approaches. Originally, the former was designed for Japanese and the latter for Korean. Further differences between the two approaches first lies in the linguistic formulation they used, which leads to differences in their rule sets. Essentially, there is only *one* rule in the HPSG-based head finalization, that is the *head finalization rule* itself. The simplicity of the rule set can be attributed to the sophisticated analysis by an HPSG parser, which shows the phrase structural as well as the syntactic head. On the other hand, the rule set in [6] contains about 20 rules, in order to arrange the position of a head word with its modifiers. It can be observed that a good HPSG parser is required for [4] if we want to expand the approach to more source-side languages, despite the simple rule. While a dependency parser is available for more languages, the rule set in [6] is dependent on the part-of-speech (POS) tag set and dependency arc label set of the dependency parser used. The approach used in our experiments combines the simplicities of the two previous approaches. We use dependency parsers to conduct the head finalization alone without touching the arrangement of various types of modifiers.

There are also statistical pre-ordering approaches. The work of [8, 9] are early syntax-oriented approaches, that they introduce separate reordering modules into SMT systems. Recently, the approach in [10] learns pre-ordering automatically from an aligned corpus. This approach achieves nearly same performance as the rule-based approach of [6] (Table 4 in [10]). In [7], a method to learn a discriminative parser for pre-ordering from an aligned parallel training corpus is proposed. The approach takes the derivation tree as a latent variable and trains a model to maximize reordering measures. The approach is fully unsupervised but needs high quality training data (i.e., a word-aligned parallel corpus). In

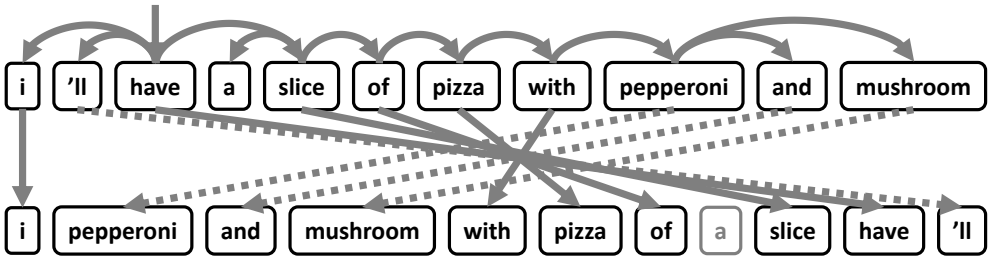


Figure 2: *Pre-ordering example of English sentence "i 'll have a slice of pizza with pepperoni and mushroom".*

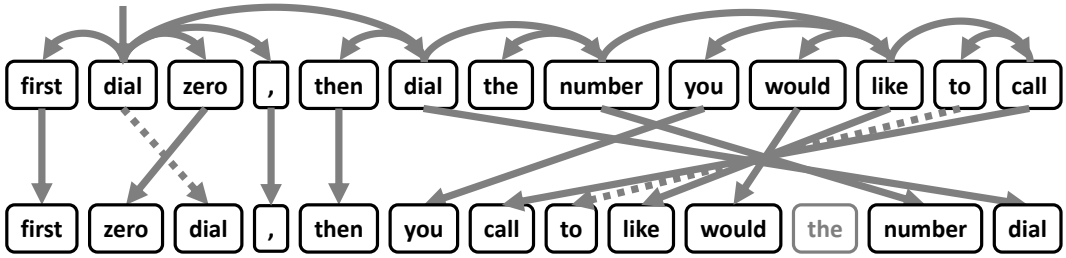


Figure 3: *Pre-ordering example of English sentence "first dial zero , then dial the number you would like to call".*

the experiments reported in [7], they show the model trained by a manually aligned parallel corpus outperforms the model trained by an automatically aligned parallel corpus of more than ten times the size. We take the approach of [7] as a baseline in our experiment, to explore the different characteristics of the rule-based and statistical pre-ordering approaches.

4. Head Finalization for Myanmar

4.1. Basic Principle

The dependency-based head finalization used in our experiment is according the following principle.

- To move the head word after all its modifiers, but
  1. do not break a coordination structure;
  2. do not cross a punctuation mark;
  3. auxiliary verbs come after their head verb.

We show two examples of the English sentence in Fig. 2 and Fig. 3. The dependency structures are marked by arcs over the words. In Fig. 2, "*pepperoni and mushroom*" is a coordination structure with the first word "*pepperoni*" as a head word. We do not apply head finalization in this kind of structure in order to keep the original order of coordinating components. In Fig. 3, the root word of the sentence, i.e. the first *dial*, does not cross the comma after it. We disable head finalization in this situation to avoid excess reordering between clauses.

As to the auxiliary verbs (3), many widely-used dependency parsers handle this kind of functional word as the modifier of a verb, just as an article becoming the modifier of a noun. While we consider auxiliary verb should be the head of a verb, and actually, in typical head-final languages the auxiliary verbs are always placed after the verb. So we arrange auxiliary verbs after their head verb. E.g., in Fig. 2, we keep the "*'ll*" after the verb "*have*"; and in Fig. 3, "*to* " after "*call*".

We describe detailed source-language dependent features in the appendices.

4.2. Myanmar Oriented Process

In the original head finalization approach, a morpheme generation process is used also to generate certain target-side grammatical markers which are absent in the source-side language. Specifically, in [4], they insert three types of tag for *topic marker*, *nominative marker*, and *accusative marker* in the source-side English. However, this issue is not so serious for Myanmar because it has a strong tendency to omit these grammatic markers as long as no ambiguity arises[2]. So we do not apply this generation process in [4] in our approach.

On the other hand, *negation* in Myanmar, unlike in Japanese or in Korean, where it is realized by a negation auxiliary word as a suffix of the verb, is realized by a prefix "မ" before the verb[3]. Further, as a collocation of the negation prefix, a negation suffix "ဘူး" must succeed the verb. Finally, the prefix and suffix surround a verb to form a negation. The phenomenon is rather like the "*ne ... pas*" in French. However, the "*pas*" is not fixed and can be replace with "*plus*"

[2]Actually, the example of a Myanmar sentence given in Fig. 1 is a quite formal expression which is rare in daily communication. We show it mainly to illustrate the syntactical similarity to Japanese and Korean.

[3]In Korean, there are also alternative prefixes used instead of negation suffixes. While in Myanmar, the negation prefix is used consistently.

or "*jamais*" and so forth according the meaning in French, while the prefix and suffix are fixed in Myanmar. We use a *neg* tag for the negation suffix generation. Specifically, the negation word of a verb is placed immediately before the verb and the *neg* tag is inserted immediately after the verb.

We use the same strategy as [4] to delete the articles in the source-side language (if any). As shown in Fig. 2 and Fig. 3, the "*a*" and "*the*" are deleted (marked in gray).

5. Experiments

5.1. Corpus and Settings

We use *Basic Travel Expression Corpus* (BTEC) [11] in the experiments. The source languages are Chinese (zh), English (en), French (fr) and the target language is Myanmar (my). The corpus statistics are is shown in Tables 1, 2 and 3. Specifically, the training, development, and test data for zh-, en-, and fr-my translations contain identical Myanmar sentences. We use two segmentation schemes for the morpheme process of Myanmar sentences. One is syllable-based (syl) [12] and the other is maximum marching (mmx) based on a dictionary with more than $20,000$ Myanmar lexicon entries. The token numbers of the two schemes are listed in the my rows in the tables (syl / mmx). Due to multi-syllable tokens, the syl has larger token numbers than mmx. We show a simple segmentation example in Fig. 4. [4]

Table 1: *Training Corpus.*

Lang.	Sentences	Tokens (syl / mmx for my)
my	$155,121$	$1,835,687$ / $1,508,234$
zh	$155,121$	$1,062,809$
en	$155,121$	$1,161,283$
fr	$155,121$	$1,248,764$

Table 2: *Development Data.*

Lang.	Sentences	Tokens (syl / mmx for my)
my	$5,000$	$59,058$ / $48,546$
zh	$5,000$	$34,103$
en	$5,000$	$37,496$
fr	$5,000$	$40,256$

Table 3: *Test Data.*

Lang.	Sentences	Tokens (syl / mmx for my)
my	$2,000$	$23,661$ / $19,425$
zh	$2,000$	$13,799$
en	$2,000$	$15,146$
fr	$2,000$	$16,173$

[4] As we have mentioned, original Myanmar morphemes are monosyllabic and there are polysyllabic morphemes of loanwords. Actually, "*word*" is not a clear (and natural) unit in Myanmar sentence. In mmx scheme, we have polysyllabic words not only derived from polysyllabic morphemes, but also derived from fixed patterns of monosyllabic morphemes, as Fig. 4 shows.

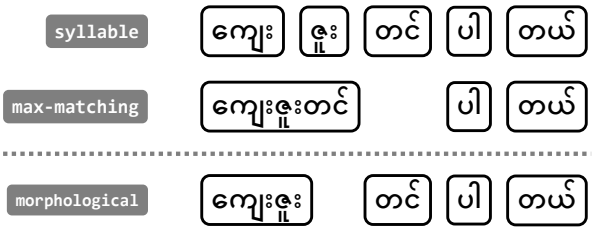


Figure 4: *Segmentation example of a Myanmar expression, meaning "thank you". The two upper rows are the syllable-based segmentation, where each box contains a syllable, and dictionary-based maximum matching, where the first three syllables are merged. The lower row illustrates a morphologically oriented analysis, where the first two syllables should be merged. The meanings of four boxes in the lower row are approximately: "gratitude", "put", polite marker, and sentence-ending marker.*

For the source-side language parsing, we use the Stanford dependency parser[5] for Chinese and English parsing [13, 14]. We use the Stanford tagger[6] [15] for French tagging (*CC* tag set [16]) and Malt parser[7] [17] for French parsing. LADER[8] is used to realize the unsupervised approach in [7] as a comparison approach. For the model training in LADER, we randomly sample $1,000$ automatically aligned sentence pairs from training set because we do not have manually-aligned data. Table 4 of [7] shows that increasing the training data for LADER from 600 to $10,000$ automatically aligned sentence pairs only brought a gain of $0.1 - 0.2$ BLEU, therefore we considered a training set size of $1,000$ to be sufficient[9].

We use the phrase-based (PB) SMT system in Moses[10] [2] as a baseline system. GIZA++[11] [18] is used to align word and alignment is symmetrized by *grow-diag-final-and* heuristics [1]. The lexicalized reordering model is trained with the *msd-bidirectional-fe* option [19]. The maximum phrase length is 7. We use SRILM[12] [20] to training 5-gram language model with interpolated modified Kneser-Ney discounting [21] on Myanmar training data.

In decoding, we adopt the default settings of the Moses decoder except the *distortion-limit* (DL). That is, *ttable-limit* is 20 and *stack* is 200. We use DL of 0, 6, 12, and ∞ in the experiments to analyze the reordering abilities of the preordering and the SMT reordering. We tuned the parameter weights on the development sets by MERT [22] and evaluated the translation on test sets by using two automatic measures: BLEU [23] and RIBES [24]. Identical decoding settings were applied on both development sets and test sets.

[5] http://nlp.stanford.edu/software/lex-parser.shtml

[6] http://nlp.stanford.edu/software/tagger.shtml

[7] http://www.maltparser.org/index.html

[8] http://www.phontron.com/lader/

[9] The training of LADER usually takes long time. Under the default settings of LADER, 500 iterations on $1,000$ sentences with 32 threads took more than 10 hours for each translation task in our experiment.

[10] http://www.statmt.org/moses/

[11] https://code.google.com/p/giza-pp/

[12] http://www.speech.sri.com/projects/srilm/

Table 4: *Test set BLEU / RIBES of* `zh-my.syl`.

DL	Baseline	LADER	Head Final.
0	35.5 / .817	36.2 / .816	38.5 / .835
6	37.9 / .831	37.9 / .830	38.7 / .832
12	38.5 / .832	37.9 / .830	38.8 / .832
∞	38.4 / .834	38.3 / .831	38.6 / .832

Table 5: *Test set BLEU / RIBES of* `en-my.syl`.

DL	Baseline	LADER	Head Final.
0	40.4 / .789	47.8 / .861	47.8 / .870
6	45.7 / .842	49.2 / .874	49.9 / .885
12	48.8 / .873	49.6 / .878	**50.3** / .886
∞	49.3 / .875	49.6 / .877	50.2 / .882

Table 6: *Test set BLEU / RIBES of* `fr-my.syl`.

DL	Baseline	LADER	Head Final.
0	36.8 / .786	43.9 / .852	43.7 / .850
6	40.9 / .825	45.2 / .859	45.6 / .860
12	45.1 / .861	45.5 / .859	**46.5** / .866
∞	45.7 / .862	45.7 / .857	**46.5** / .860

Table 7: *Test set BLEU / RIBES of* `zh-my.mmx`.

DL	Baseline	LADER	Head Final.
0	32.9 / .799	34.6 / .810	35.4 / .811
6	34.9 / .816	35.1 / .816	**36.5** / .821
12	35.5 / .817	35.7 / .817	**36.5** / .819
∞	35.2 / .816	35.6 / .814	**36.5** / .820

Table 8: *Test set BLEU / RIBES of* `en-my.mmx`.

DL	Baseline	LADER	Head Final.
0	40.4 / .802	48.0 / .867	47.8 / .871
6	44.7 / .835	48.9 / .871	49.0 / .881
12	48.6 / .873	49.5 / .877	49.8 / .880
∞	49.0 / .876	49.5 / .875	49.7 / .878

Table 9: *Test set BLEU / RIBES of* `fr-my.mmx`.

DL	Baseline	LADER	Head Final.
0	36.9 / .791	43.6 / .844	43.6 / .847
6	39.7 / .818	44.7 / .852	44.9 / .855
12	44.3 / .855	45.1 / .852	45.4 / .855
∞	44.7 / .856	45.4 / .853	45.3 / .853

5.2. Results

We list the experimental results of three source languages (`zh`, `en`, `fr`) with two target Myanmar segmentation schemes (`my.syl`, `my.mmx`) in Tables 4 – 12. In each table, two evaluation measures (BLEU / RIBES) are given with different distortion limits (DLs). The best BLEU scores among the different DLs are underlined and bold BLEU scores are significantly different ($p < 0.05$) to the best baseline BLEU score. As the log-linear model weights were tuned to optimize the BLEU rather than the RIBES score on the development sets with MERT, the RIBES scores shown in the tables are only a complementary evaluation of translation performance on word order.

Table 10: *Test set BLEU / RIBES on* `syl` *of* `zh-my.mmx`.

DL	Baseline	LADER	Head Final.
0	36.8 / .818	38.0 / .829	38.5 / .829
6	38.4 / .836	39.0 / .835	39.7 / .838
12	39.0 / .837	39.4 / .835	**39.9** / .838
∞	38.6 / .833	39.2 / .832	39.6 / .838

Table 11: *Test set BLEU / RIBES on* `syl` *of* `en-my.mmx`.

DL	Baseline	LADER	Head Final.
0	45.0 / .814	51.2 / .879	51.5 / .882
6	48.6 / .847	52.1 / .883	52.7 / .891
12	52.0 / .882	52.8 / .887	**53.4** / .890
∞	52.5 / .885	52.8 / .887	53.2 / .889

Table 12: *Test set BLEU / RIBES on* `syl` *of* `fr-my.mmx`.

DL	Baseline	LADER	Head Final.
0	40.5 / .803	47.0 / .857	46.6 / .860
6	43.1 / .831	48.0 / .865	47.9 / .869
12	47.5 / .867	48.0 / .864	48.4 / .870
∞	47.8 / .867	48.4 / .865	48.3 / .865

In Tables 4 – 6, the evaluation is on `syl` and in Tables 7 – 9, on `mmx`. So the results in the corresponding tables of these two groups are not comparable. In Tables 10 – 12, we show the results on `syl` for `mmx` outputs. So, the corresponding results in Tables 4 – 6 and Tables 10 – 12 are comparable.

5.3. Discussion

In Tables 1 – 3, it can be observed that the average sentence length of the corpus used is quite small (all less than 10 except for `my.syl`). This is because the corpus mainly contains colloquial, rather than literary sentences. This bias suggests two problems. First, the state-of-the-art Moses system can handle the reordering well for short sentences, where a pre-ordering approach may not show its power. Second, there may be more errors in parsing colloquial sentences than literary ones, which may reduce the performance of rule-based head finalization.

Using the same analysis as in [4], first we calculate the average *Kendall's* τ on the training sets (Table 13) to investigate the reordering performance. We observed the following phenomena:

Table 13: *Average Kendall's τ on training sets.*

Language Pair	Baseline	LADER	Head Final.
zh–my.syl	.69	.79	.83
en–my.syl	.53	.79	.79
fr–my.syl	.53	.76	.75
zh–my.mmx	.69	.80	.83
en–my.mmx	.53	.79	.79
fr–my.mmx	.54	.76	.76

- The two different segmentation schemes of Myanmar lead to very similar average Kendell's τ.
- LADER can produce an average Kendall's τ of around .75 – .80 irrespective of the value of average Kendall's τ in its input corpus.
- Dependency-based head finalization shows identical performance to LADER in en–my and fr–my, but better performance on zh–my, where the corpus before pre-ordering already has a relatively high average Kendall's τ.

From Table 13, it is noticeable that en–my and fr–my have nearly identical characteristics while zh–my is different from them. This phenomenon is reflected in the evaluation results on the test sets.

In zh–my translation, we find LADER hardly improves performance over the baseline SMT system in both syl and mmx, while the head finalization approach improves performance over the baseline in both cases and more substantially for mmx. LADER has higher performance on en–my and fr–my, and the proposed head finalization technique has identical or better performance. Since the difference in word order is not as severe for zh–my as for en–my and fr–my (as indicated by the Kendall's τ statistics), we consider rule-based head finalization to be a better complementary approach for the SMT system for zh–my. For language pairs with considerably different word orders as en–my and fr–my, LADER and rule-based head finalization, despite their essentially different mechanisms, attain similar levels of performance.

In Tables 4 – 12, it can also be noticed that the differences are quite large between DL = 0 (i.e. monotone translation) and the corresponding best BLEU in each baseline result, but the differences are reduced by both pre-ordering approaches. So, the performance gains over the baseline by using pre-ordering diminish as the DL is increased. As to the RIBES score, the differences actually are not substantial between the baseline, LADER, and the head finalization approach. We consider these to be reasonable phenomena caused by the short length of the sentences in the corpus.

A major factor affecting the performance of the rule-based head finalization approach is the precision of the parser used, and perhaps the most important factor affecting the performance of a statistical approach, such as LADER, is the quality of the training data. In the survey conducted in [25], they reported "*we observed relatively small effects on reordering quality in response of parsing errors*". We visually inspected a sample of the parsing results used in our experiments and found parsing errors did not have a large effect on the performance of our head finalization approach. We consider a major benefit of our approach is that we almost always use the "*head*" information from a dependency parse, which leads to robustness. The performance of LADER is greatly affected by the quality rather than the amount. So it is sensitive to the nature of languages involved, and also to their word segmentation schemes because they affect the quality of word alignment used to train LADER.

Among the various segmentation schemes for Myanmar, we believe the syl strategy has a tendency to over-split sentences and mmx may lead to some long expressions without necessary splits as illustrated in Figure 4. It can be seen that the data segmented using mmx has fewer tokens and relatively longer words. It was expected that the the evaluation scores in Table 7 – 9 would be lower than those in Table 4 – 6. Conversely, if the translation is done on mmx and evaluated by syl, as shown in Table 10 – 12, we find the results are better than those in Table 4 – 6. The experimental results show that the mmx strategy is a better segmentation strategy than syl. Although mmx introduces long expressions, it can offer more meaningful units in word alignment and translation, which lead to a better performance. However, a more useful standard morpheme analysis system should hopefully be built for Myanmar in the future.

We show translation examples of zh–my, en–my and fr–my. The examples are selected from the best results of mmx and illustrated using syl segmentation. It can be seen that the head finalization has a rigidity with respect to the syntactic structure. For example, the objects of verbs are strictly arranged in front positions in head finalization (actually, untouched), such as the Chinese "我" in Fig. 5, the English "*i*" in Fig. 6, and the French "*j'* " in Fig. 7. While in the pre-ordering from LADER, those words are scattered. For example, in the first example of Fig. 6 and in Fig. 7, the "*i*" and "*j'* " are moved to the end of the sentences. This is because LADER does not have information on the syntactic structure of a sentence. In this example, LADER moves the phrases "*i want*" and "*j' ai*" as whole units to the sentence ends, and makes further local swapping within the phrases. The second example of Fig. 6 shows the simplicity of our head finalization approach; in this example, only the verb "*bring*" is moved to the end of the sentence.

6. Conclusion and Future Work

In this paper, we conducted pre-ordering experiments on Chinese-, English-, French-to-Myanmar translation. We found that a simple dependency-based head finalization pre-ordering strategy can consistently and efficiently improve a baseline SMT system. The proposed head finalization approach does not require parallel training data, and only de-

Baseline Input	对不起， 请 告诉 我 这个 怎么 用 ？ *(sorry , please tell me this how use ?)*
LADER Input	这个 对不起， 请 告诉 我 怎么 用 ？ *(this sorry , please tell me how use ?)*
Head Final. Input	对不起， 我 这个 怎么 用 告诉 请 ？ *(sorry , me this how use tell please ?)*
Baseline Output	တ ဆိတ် လောက် ၊ ကျေး ဇူး ပြု ပြီး ဒီ ဟာ ကို ဘယ် လို သုံး ရ မ လဲ ။
LADER Output	ကျေး ဇူး ပြု ပြီး တ ဆိတ် လောက် ။ ကျေး ဇူး ပြု ပြီး ဒီ ဟာ ကို ဘယ် လို သုံး ရ မ လဲ ။
Head Final. Output	တ ဆိတ် လောက် ၊ ဒါ ကို ဘယ် လို သုံး ရ မ လဲ ဆို တာ ပြော ပြ ပေး နိုင် မ လား ။
Reference	တ ဆိတ် လောက် ၊ ဒါ ကို ဘယ် လို သုံး ရ တယ် ဆို တာ ပြ ပေး ပါ လား ။

Figure 5: *Chinese-to-Myanmar translation example. (For the input rows, word-by-word English literal translations are annotated in gray. An unconstrained translation of the original Chinese sentence is "Excuse me, could you tell me how this works?")*

Baseline Input	**i want to send this parcel to japan .**	**please bring me some ice .**
LADER Input	**this parcel to japan send to want i .**	**ice some me please bring .**
Head Final. Input	**i this parcel japan to send to want .**	**please me some ice bring .**
Baseline Output	ဒီ အ ထုတ် ကို ဂျ ပန် ကို ပို့ ချင် လို့ ပါ ။	ရေ ခဲ ကို နည်း နည်း လောက် ယူ လာ ပေး ပါ ။
LADER Output	ဒီ ပါ ဆယ် ဂျ ပန် ကို ပို့ ချင် ပါ တယ် ။	ရေ ခဲ နည်း နည်း ယူ လာ ပေး ပါ နော် ။
Head Final. Output	ဒီ ပါ ဆယ် ဂျ ပန် ကို ပို့ ချင် လို့ ပါ ။	ရေ ခဲ ယူ လာ ပေး ပါ ။
Reference	ဒီ ပါ ဆယ် ဂျ ပန် ကို ပို့ ချင် လို့ ပါ ။	ရေ ခဲ ကို ယူ လာ ပေး ပါ ။

Figure 6: *English-to-Myanmar translation examples.*

Baseline Input	**j' ai oublié mon billet d' avion .** *(I have forgotten my ticket of aeroplane .)*
LADER Input	**avion d' billet mon oublié ai j' .** *(aeroplane of ticket my forgetten have I .)*
Head Final. Input	**j' mon avion d' billet oublié ai .** *(I my aeroplane of ticket forgetten have .)*
Baseline Output	ကျွန် တော့် ရဲ့ လေ ယာဉ် လက် မှတ် မေ့ ကျန် ခဲ့ တယ် ။
LADER Output	လေ ယာဉ် လက် မှတ် မေ့ ကျန် ခဲ့ တယ် ။
Head Final. Output	လေ ယာဉ် လက် မှတ် မေ့ ကျန် ခဲ့ ပါ တယ် ။
Reference	လေ ယာဉ် လက် မှတ် မေ့ ကျန် ခဲ့ ပါ တယ် ။

Figure 7: *French-to-Myanmar translation example. (For the input rows, word-by-word English literal translations are annotated in gray. An unconstrained translation of the original French sentence is "I forgot my airline ticket.")*

pends on a source-side dependency parser, which allowed it to attain higher performance than an unsupervised baseline in our experiment. The simplicity and efficiency of the proposed head finalization approach should allow it to find practical application on large scale data sets.

In further work, we plan to expand the parallel data and conduct experiments on larger corpora. We are also developing a morpheme analyzer and parsers for Myanmar to facilitate the transference of more techniques of Japanese and Korean language processing to Myanmar language processing.

7. References

[1] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation." in *Proc. of HTL-NAACL*, 2003, pp. 48–54.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation." in *Proc. of ACL*, 2007, pp. 177–180.

[3] P. Koehn, "Europarl: A parallel corpus for statistical machine translation." in *Proc. of MT summit*, 2005, pp. 79–86.

[4] H. Isozaki, K. Sudoh, H. Tsukada, and K. Duh, "HPSG-based preprocessing for English-to-Japanese translation." *ACM Transactions on Asian Language Information Processing*, vol. 11, no. 3, 2012, article 8.

[5] H. Dan, K. Sudoh, X. Wu, K. Duh, H. Tsukada, and M. Nagata, "Head finalization reordering for Chinese-to-Japanese machine translation." in *Proc. of SSST*, 2012, pp. 57–66.

[6] P. Xu, J. Kang, M. Ringgaard, and F. Och, "Using a de-

pendency parser to improve SMT for subject-object-verb languages." in *Proc. of HLT-NAACL*, 2009, pp. 245–253.

[7] G. Neubig, T. Watanabe, and S. Mori, "Inducing a discriminative parser to optimize machine translation reordering." in *Proc. of EMNLP-CoNLL*, 2012, pp. 843–853.

[8] F. Xia and M. McCord, "Improving a statistical mt system with automatically learned rewrite patterns," in *Proc. of COLING*, no. 508, 2004.

[9] M. Collins, P. Koehn, and I. Kučerová, "Clause restructuring for statistical machine translation," in *Proc. of ACL*, 2005, pp. 531–540.

[10] D. Genzel, "Automatically learning source-side reordering rules for large scale machine translation." in *Proc. of COLING*, 2010, pp. 376–384.

[11] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, "Creating corpora for speech-to-speech translation." in *Proc. of EUROSPEECH*, 2003, pp. 381–384.

[12] Y. K. Thu, A. Finch, Y. Sagisaka, and E. Sumita, "A study of Myanmar word segmentation schemes for statistical machine translation," in *Proc. of ICCA*, 2013, pp. 167–179.

[13] R. Levy and C. Manning, "Is it harder to parse Chinese, or the Chinese treebank?" in *Proc. of ACL*, 2003, pp. 439–446.

[14] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars." in *Proc. of ACL*, 2013, pp. 455–465.

[15] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network." in *Proc. of HLT-NAACL*, 2003, pp. 173–180.

[16] B. Crabbé and M. Candito, "Expériences d'analyse syntaxique statistique du français." in *Proc. of TALN*, 2008.

[17] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi, "MaltParser: A language-independent system for data-driven dependency parsing." *Natural Language Engineering*, vol. 13, no. 02, pp. 95–135, 2007.

[18] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models." *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[19] C. Tillmann, "A unigram orientation model for statistical machine translation." in *Proc. of HTL-NAACL*, 2004, pp. 101–104.

[20] A. Stolcke, "SRILM–an extensible language modeling toolkit." in *Proc. of ICSLP 2002*, 2002, pp. 901–904.

[21] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling." TR-10-98, Computer Science Group, Harvard Univ., Tech. Rep., 1998.

[22] F. J. Och, "Minimum error rate training in statistical machine translation." in *Proc. of ACL*, 2003, pp. 160–167.

[23] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation." in *Proc. of ACL*, 2002, pp. 311–318.

[24] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada, "Automatic evaluation of translation quality for distant language pairs." in *Proc. of EMNLP*, 2010, pp. 944–952.

[25] D. Han, P. Martınez-Gómez, Y. Miyao, K. Sudoh, and M. Nagata, "Effects of parsing errors on pre-reordering performance for Chinese-to-Japanese SMT," in *Proc. of PACLIC*, 2013, pp. 267–276.

A. Head Finalization for Chinese

We use the Standford Chinese dependency parser.

- The `conj` arc is used to identify coordination.
- The `punct` arc is used to identify punctuation marks.
- The `asp`, `assm`, `ba`, `cop`, `cpm`, `dvpm`, `mmod` arcs are taken as auxiliary verbs or post-positioned particles. They are always arranged after their heads.
- The `neg` arc is used to identify the negation.
- We clean up parsing errors around several common Chinese function words, to insure:
  - sentence final particles "啊", "吧", "的", "了", "吗", "呢", "呀" are always after their head words;
  - determiners "这", "那", "哪" are always before their head words.
- The article deletion process is not applied in Chinese.

B. Head Finalization for English

We use the Standford English dependency parser.

- The `conj`, `cc` arcs are used to identify coordination.
- The `punct` arc is used to identify punctuation marks.
- The `aux`, `auxpass`, `cop` arcs are taken as auxiliary verbs. They are always arranged after their heads.
- The `mark` arc and "*when*", "*where*" with `advmod` arc are always arranged after their heads.
- The `neg` arc is used to identify the negation.
- The "*there be*" of an existential clause is kept together.
- For the process of article deletion, we delete "*a*", "*an*", "*the*".

C. Head Finalization for French

We use Malt French parser with the CC tag set.

- The `*coord*` arcs are used to identify coordination.
- The `ponct` arc is used to identify punctuation marks.
- The `*aux*` arcs are taken as auxiliary verbs. They are always arranged after their heads.
- The "*ne*", "*n'* " with `mod` arc is used to identify the negation.
- The "*il y a*" and "*y a-t-il*" of an existential clause is kept together.
- For the process of article deletion, we delete "*le*", "*la*", "*l'* ", "*les*", "*un*", "*une*".

Discriminative Adaptation of Continuous Space Translation Models

Quoc-Khanh Do[1,2]*, Alexandre Allauzen*[1,2]*, François Yvon*[1]

LIMSI-CNRS[1] and Univ. Paris-Sud[2], rue John von Neumann, F 91 403 Orsay

{dokhanh,allauzen,yvon}@limsi.fr

Abstract

In this paper we explore various adaptation techniques for continuous space translation models (CSTMs). We consider the following practical situation: given a large scale, state-of-the-art SMT system containing a CSTM, the task is to adapt the CSTM to a new domain using a (relatively) small in-domain parallel corpus. Our method relies on the definition of a new discriminative loss function for the CSTM that borrows from both the max-margin and pair-wise ranking approaches. In our experiments, the baseline out-of-domain SMT system is initially trained for the WMT News translation task, and the CSTM is to be adapted to the lecture translation task as defined by IWSLT evaluation campaign. Experimental results show that an improvement of 1.5 BLEU points can be achieved with the proposed adaptation method.

1. Introduction

Domain adaptation (DA) is an important and active research topic in Statistical Natural Language Processing [1, 2]. In a nutshell, domain adaptation aims to mitigate the well-known problem of *covariate shift* which stems from statistical distribution differences between train and test samples. This often happens in NLP, especially when train and test documents correspond to different genres, registers or domains. Domain adaptation is often expressed in terms of finding an optimal combination of a small in-domain dataset with large amounts of out-of-domain data.

To avoid the dilution of domain-specific knowledge, most approaches consider various kinds of data weighting schemes in order to balance the importance of in-domain *vs* out-of-domain data. In such adaptation scenarios, the NLP component needs to be retrained, entirely or partly, to integrate these new samples, which can be very time consuming or even unrealistic in many situations. This is especially problematic for SMT systems, that are typically made of several layers of statistical models. DA for SMT has therefore received considerable attention in the recent years (for instance [3, 4, 5, 6]). This situation is compounded when, as we do here, SMT systems rely on Continuous Space Language Models (CSLMs) or Translation Models (CSTMs), which have recently gained a lot of popularity [7, 8, 9, 10, 11, 12].

As demonstrated for many NLP tasks [13], such as language modelling [7, 14, 15, 16], syntactic parsing [17] and machine translation [8, 9, 18, 19], CSLMs and CSTMs can remedy to two well-know issues of statistical modelling for linguistic data. Typical statistical models use discrete random variables to represent the realization of words, phrases or phrase pairs. The corresponding parameter estimates are based on relative frequencies and are unreliable for rare events. Furthermore, the resulting representations ignore morphological, syntactic and semantic relationships that exist among linguistic units. This lack of structure hinders the generalization power of statistical models and reduces their ability to adapt to other domains. By contrast, continuous models manipulate numerical representations of linguistic units that are automatically trained from large corpora and that implicitly capture some similarity relationships, thereby introducing some smoothing in the probability estimates.

The adaptation of Continuous Models for SMT has thus far received little attention. We study here the following practical situation: a large scale, state-of-the-art SMT system is available and needs to be ported to a new domain, using a small in-domain parallel corpus. In this setting, our main contribution is the definition and evaluation of new loss functions, that aim at discriminatively adapting the CSTMs to the new data. These objective functions derive from both the max-margin [20, 21] and pair-wise ranking [22, 23] approaches. In our experiments, the baseline, out-of-domain system is preliminarily trained for the News translation task, and the CSTMs must be adapted to the lecture translation task as defined in recent IWSLT evaluation campaigns [24].

The rest of the paper is organized as follows. Section 2 briefly describes the model structure that will be used in our experiments. Section 3 proposes new discriminative loss functions on N-best lists, along with the corresponding adaptation algorithms. The next section gives details about our experimental conditions and analyzes our main results. We finally provide a short review of similar works both on Discriminative Machine Translation and on Continuous Space Translation Models, before concluding with some perspectives for future work.

2. Continuous space translation models

This section provides an overview of the CSTM used in our baseline system and subsequently adapted. This model was introduced and fully described in [9].

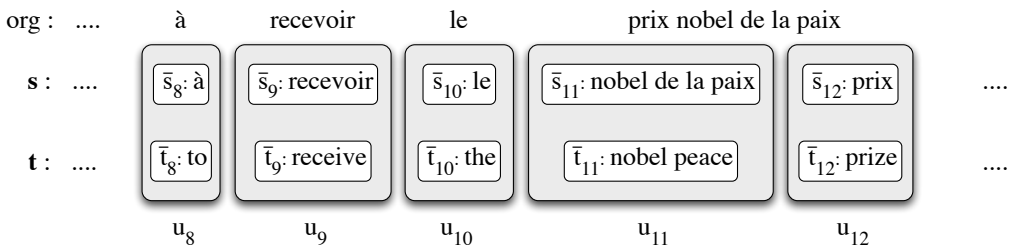


Figure 1: Extract of a French-English sentence pair segmented in bilingual units. The original (*org*) French sentence appears at the top of the figure, just above the reordered source $\mathbf{s}$ and target $\mathbf{t}$. The pair $(\mathbf{s}, \mathbf{t})$ decomposes into a sequence of L bilingual units (*tuples*) $\mathrm{u}_1, ..., \mathrm{u}_L$. Each tuple u_i contains a source and a target phrase: $\bar{s}_i$ and $\bar{t}_i$.

2.1. The n-gram translation model

n-gram translation models (TMs) rely on a specific decomposition of the joint probability $P(\mathbf{s}, \mathbf{t})$, where $\mathbf{s}$ is a sequence of I *reordered* source words $(s_1, ..., s_I)$[1] and $\mathbf{t}$ contains J target words $(t_1, ..., t_J)$. This sentence pair is assumed to be decomposed into a sequence of L bilingual units called *tuples* defining a joint segmentation: $(\mathbf{s}, \mathbf{t}) = (\mathrm{u}_1, ..., \mathrm{u}_L)$. In this framework, the basic translation units are *tuples*, which are the analogous of phrase pairs, and represent a matching $\mathrm{u} = (\bar{s}, \bar{t})$ between a source $\bar{s}$ and a target $\bar{t}$ phrase (Figure 1). Using the n-gram assumption, the joint probability of a *synchronized* and *segmented* sentence pair is:

$$P(\mathbf{s}, \mathbf{t}) = \prod_{i=1}^{L} P(\mathrm{u}_i | \mathrm{u}_{i-n+1}^{i-1}), \quad (1)$$

where u_{i-n+1}^{i-1} denotes the tuple sequence $\mathrm{u}_{i-n+1}, \ldots, \mathrm{u}_{i-1}$. The complete model for a sentence pair thus involves latent variables that specify the reordering applied to the source sentence, as well as its segmentation into translation units. These latent variables define the derivation of the source sentence that generates the target sentence. They are omitted for the sake of clarity. During the training step, the segmentation is a by-product of source reordering, and ultimately derives from initial word and phrase alignments (see [25, 26] for details). During the inference step, the SMT decoder will compute and output the best derivation.

In this model, the elementary units are bilingual pairs, which means that the underlying vocabulary, hence the number of parameters, can be quite large, even for small translation tasks. Due to data sparsity issues, such models face severe estimation problems. Equation (1) can therefore be factored by decomposing tuples in two (source and target) parts and in two equivalent ways:

$$\begin{aligned} & P(\mathrm{u}_i | \mathrm{u}_{i-n+1}^{i-1}) \\ & \quad = P(\bar{t}_i | \bar{s}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1}) P(\bar{s}_i | \bar{s}_{i-n+1}^{i-1}, \bar{t}_{i-n+1}^{i-1}) \quad (2) \\ & \quad = P(\bar{s}_i | \bar{t}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1}) P(\bar{t}_i | \bar{s}_{i-n+1}^{i-1}, \bar{t}_{i-n+1}^{i-1}) \end{aligned}$$

[1] In the context of the n-gram translation model, $(\mathbf{s}, \mathbf{t})$ thus denotes an *aligned* sentence pair, where the source words are reordered.

Each decomposition involves two bilingual conditional distributions that can also be decomposed at the level of words, using again the n-gram assumption.

2.2. Continuous translation modeling with SOUL

The n-gram distributions described in Section 2.1 are defined over potentially large vocabularies. As proposed in [9], these distributions can be estimated using the SOUL model introduced in [27]. Following [28], the SOUL model combines the feed-forward neural network approach for n-gram models [7] with a class-based prediction [29]. Structuring the output layer with word-class information makes the estimation of distributions over the entire vocabulary computationally feasible. Neural network architectures are also interesting as they can easily handle larger contexts than typical n-gram models. In the SOUL architecture, enlarging the context mainly consists in increasing the size of the projection layer, which corresponds to a simple look-up operation. Increasing the context length at the input layer thus causes only a linear growth in complexity in the worst case [14].

2.3. Training and initialization issues

The word-based translation model described in section 2.1 involves two different languages and thus two different vocabularies: the predicted unit is a target or source word, whereas the context is made of both source and target words. As proposed in [9], the SOUL architecture is modified to make up for *mixed* contexts by considering two different sets of word embeddings, one for each language. Training this kind of model can be achieved by maximizing the log-likelihood on some parallel corpus. Following [9], this optimization is performed by stochastic back-propagation, while the derivation (source reordering and segmentation in translation units) are derived by the usual procedure (see [30]).

However, for multi-layered neural networks, the non-convexity of the objective function implies that the parameter initialization can highly impact the training process in terms of its convergence speed and of its performance. In the bilingual context of translation modeling, two monolingual language models can first be estimated for initialization

purpose[2]. In a domain adaptation context, we assume that an existing CSTM –trained on the out-of-domain data– already exists. This model is thus well suited to bootstrap the adaptation process.

3. Objective functions for adaptation

In most previous works (eg. [8, 9]), CSTMs are estimated by maximizing the regularized conditional log-likelihood (CLL) on parallel training corpora. This estimation procedure is used to train a baseline CSTM on the out-of-domain corpus, producing a baseline model that will serve as an initial point for domain adaptation. Given a small in-domain parallel corpus, the same training procedure can also be used. A straightforward adaptation algorithm consists in running a few epochs of the standard back-propagation algorithm on the in-domain data to maximize the conditional likelihood using, as initial parameters, the out-of-domain model.

There is however only a loose relationship between the CLL criterion and the final translation quality. The CSTM is usually integrated in the translation process through a reranking step, the goal of which is to reorder a reduced set of candidate translations, called N-best list. Therefore, to better take advantage of the small amount of in-domain data, we propose to explore alternative objective functions that are more directly related to the translation quality (as reflected by the BLEU score) after reranking. We first present the general learning algorithm, then the various objective functions.

3.1. Rescoring N-best lists with CSTMs

Due to the high computational cost of normalizing the output layer, continuous models are in most cases[3] introduced in a post-processing step called N-best reranking.

We thus assume that for each source sentence $\mathbf{s}$, the decoder can generate an N-best list $\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N\}$ of N top translation candidates. Each hypothesis $\mathbf{h}_i = (\mathbf{t}_i, \mathbf{a}_i)$ is associated with the decoder score $F_{\boldsymbol{\lambda}}(\mathbf{s}, \mathbf{h})$ computed as:

$$F_{\boldsymbol{\lambda}}(\mathbf{s}, \mathbf{h}) = \sum_{k=1}^{K} \lambda_k f_k(\mathbf{s}, \mathbf{h}), \tag{3}$$

where K feature functions (f_k) are weighted by a set of coefficients (λ_k). The n-gram approach differs from other approaches by the hidden variables associated to derivations, such as the source word reordering and the segmentation of the resulting parallel sentence. The basic feature functions used in this study are very similar to those used by standard phrase-based SMT systems (see [30] for instance).

When reranking with a continuous space model, $F_{\boldsymbol{\lambda}}(.)$ is augmented to also include an additional feature denoted $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h})$. As explained in Section 2.2, $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h})$ typically corresponds to the negated log-probability of the derivation: $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}) = -\log P_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h})$, where $\boldsymbol{\theta}$ is the vector containing the CSTM's free parameters. The scoring function used in reranking is then:

$$G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}) = F_{\boldsymbol{\lambda}}(\mathbf{s}, \mathbf{h}) + \lambda_{K+1} f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}) \tag{4}$$

This scoring function depends on the CSTM's parameters $\boldsymbol{\theta}$, as well as on the coefficients $\boldsymbol{\lambda}$ of the scoring function. In the approach proposed here, optimizing the reranking step will thus requires *to alternatively tune the vector of coefficients $\boldsymbol{\lambda}$ and to adapt the CSTM's weight vector $\boldsymbol{\theta}$*: the former procedure uses the development data, while the latter will use the in-domain parallel corpus.

Algorithm 1 Joint optimization procedure for $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$

```
Initialize θ and λ
for each iteration do
    for M mini-batches do                          ▷ λ is fixed
        Compute the sub-gradient of L(θ, s) for all s in the mini-batch
        Update θ
    end for
    Update λ using dev set                         ▷ θ is fixed
end for
```

The corresponding proposed optimization procedure splits the in-domain set in mini-batches of a fixed size (typically 128 subsequent sentence pairs). As sketched in Algorithm 1, each mini-batch is used to update the parameters $\boldsymbol{\theta}$ of the CSTM while keeping $\boldsymbol{\lambda}$ fixed. The vector $\boldsymbol{\lambda}$ is updated every M mini-batches.

In our study, tuning $\boldsymbol{\lambda}$ is performed using standard tools (here, the K-Best Mira algorithm described in [21] as implemented in MOSES [4]). The training of CSTMs (with fixed $\boldsymbol{\lambda}$) is more interesting and we compare two discriminative objective functions, which aim at better taking the translation quality into account. These two objectives are in turn compared to the conventional maximization of the conditional likelihood criterion on parallel data.

3.2. A max-margin approach

As explained above, each hypothesis $\mathbf{h}_i$ produced by the decoder is scored according to (4). Its quality can also be evaluated by the sentence-level approximation of the BLEU score $sBLEU(\mathbf{h}_i)$. Let $\mathbf{h}^*$ denote the hypothesis with the best sentence BLEU score. A max-margin loss function [33, 34, 20] for estimating $\boldsymbol{\theta}$ can then be formulated as follows:

$$\mathcal{L}_{mm}(\boldsymbol{\theta}, \mathbf{s}) = -G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}^*) + \max_{1 \le j \le N} \left(G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_j) + \text{cost}_\alpha(\mathbf{h}_j) \right), \tag{5}$$

where $\text{cost}_\alpha(\mathbf{h}_j) = \alpha\big(sBLEU(\mathbf{h}^*) - sBLEU(\mathbf{h}_j)\big)$. The parameter α mitigates the contribution of the cost function

[2] The following parameters can be initialized given a source and target language monolingual models: the source and target word embeddings respectively, and the structured output layer's structure.

[3] See however [31, 32, 19] for early attempts to integrate Neural Network Translation Models within the decoder.

[4] http://www.statmt.org/moses/

to the objective function. When $alpha > 0$, the objective defined in (5) is a general max-margin training criterion; taking $\alpha = 0$ corresponds to the structured perceptron loss [35]. This objective function aims to discriminatively learn to give the highest model score to the hypothesis $\mathbf{h}^*$ having the best sentence level BLEU. Moreover, the margin term enforces the scoring difference between $\mathbf{h}^*$ and the rest of the N-best list to be greater than the margin.

However, a source sentence can have, among the N-best list, several good translations that differ only slightly from the best hypothesis. The max-margin objective function defined above nevertheless considers that all hypotheses, except the best one, are wrong. The ranking-based approach defined below tries to correct this weakness.

3.3. Pairwise ranking

Inspired by [22], we define another objective function that aims to learn the ranking of a set of hypotheses with respect to their BLEU scores. Assuming that r_i denotes the rank of the hypothesis $\mathbf{h}_i$ when the N-best list is reordered according to the sentence-level BLEU, this objective is defined as:

$$\mathcal{L}_{pro}(\boldsymbol{\theta}, \mathbf{s}) = \sum_{1 \leq i,k \leq N} \mathbb{I}_{\{r_i + \delta \leq r_k, G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}_i) < G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s},\mathbf{h}_k)\}} \left(-G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) + G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k)\right). \quad (6)$$

Note that this loss function only involves a subset of the $N(N-1)/2$ pairs of hypotheses, since two hypotheses are included in the sum only if they are sufficiently apart in terms of their ranks: formally, the absolute difference of ranks should be greater than a predefined threshold δ. As in PRO [22], the ranking problem is thus reduced to a binary classification task taking candidate translation pairs as inputs. A major difference to PRO though, is the fact that we use this loss function to train the CSTM's parameters $\boldsymbol{\theta}$, rather than the feature weights $\boldsymbol{\lambda}$.

This ranking criterion can finally be generalized again with the notion of margin: for a pair of hypotheses $(\mathbf{h}_i, \mathbf{h}_k)$ such as $r_i + \delta < r_k$, the scoring difference $G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) - G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k)$ should exceed a positive margin. As in section 3.2, the margin is based on the sentence-level BLEU score via the use of the cost function cost_α. Let us define the set of all critical pairs of hypotheses as:

$$\mathcal{C}_\delta^\alpha = \{(i,k) : 1 \leq i,k \leq N, r_i + \delta \leq r_k, \quad (7)$$
$$G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) - G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k) < \text{cost}_\alpha(\mathbf{h}_k) - \text{cost}_\alpha(\mathbf{h}_i)\}.$$

The objective function that combines both the pairwise ranking and max-margin criterion is defined as follows:

$$\mathcal{L}_{pro-mm}(\boldsymbol{\theta}, \mathbf{s}) = \sum_{(i,k) \in \mathcal{C}_\delta^\alpha} \text{cost}_\alpha(\mathbf{h}_k) - \text{cost}_\alpha(\mathbf{h}_i) - G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_i) + G_{\boldsymbol{\lambda},\boldsymbol{\theta}}(\mathbf{s}, \mathbf{h}_k). \quad (8)$$

Taking $\alpha = 0$, this function is equivalent to the pairwise ranking criterion (6).

4. Experiments

We now turn to an experimental comparison of the adaptation methods described in Section 3. In our experimental framework, the lecture translation task defines the targeted (or in) domain, while the baseline system corresponds to a state-of-the-art SMT system, intensively trained for the News translation task, as defined by the WMT evaluation. The goal is therefore to quickly and efficiently adapt this out-of-domain system by only updating the CSTM.

4.1. Task and corpora

The task considered here is derived from the text translation track of IWSLT 2011 from English to French (the TED Talks task [24]), where a (in-domain) training dataset containing $107,058$ aligned sentence pairs was made available. As explained above, this corpus only serves to adapt the continuous space translation models, *i.e* to adapt the parameters $\boldsymbol{\theta}$. The baseline and out-of-domain system is trained in the condition of the shared translation task of WMT 2013 evaluation campaign.[5] This system includes CSTMs that will be used as starting points for adaptation.

The official development and test sets respectively contain 934 and $1,664$ sentence pairs. Following [9], these sets are swapped, the tuning of the feature weights $\boldsymbol{\lambda}$ is carried out on $1,664$ sentences of the latter, while the final test is on 934 sentences of the former. Translations are evaluated using the BLEU score [36]. For a fair comparison, all BLEU scores reported are obtained after a tuning phase on the dev set, including the baseline system. For Algorithm 1, $(\boldsymbol{\theta}, \boldsymbol{\lambda})$ are selected by maximizing the BLEU score on the dev set (line 7).

4.2. Baseline system and models

The n-gram-based system used here is based on an open source implementation[6] of the bilingual n-gram approach to Statistical Machine Translation [37]. In a nutshell, the translation model is implemented as a stochastic finite-state transducer trained using an n-gram model of (source, target) pairs as described in section 2.1. Training this model requires to reorder source sentences so as to match the target word order. This is performed by a non-deterministic finite-state reordering model, which uses part-of-speech information generated by the TreeTagger to generalize reordering patterns beyond lexical regularities.

In addition to the TM, fourteen feature functions are included that are similar to the standard phrase-based system: *target-language model*; four *lexicon models*; six *lexicalized reordering models*; a distance-based *distortion model*; and finally a *word-bonus model* and a *tuple-bonus model*. A more detailed description is in [30].

[5] http://www.statmt.org/wmt13/

[6] `perso.limsi.fr/Individu/jmcrego/bincoder`

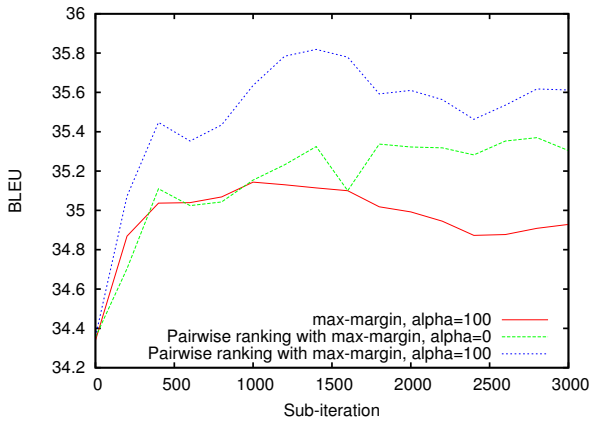


Figure 2: Evolution of BLEU scores on the dev set using three discriminative criteria described in (5), (6) and (8). Vector $\boldsymbol{\lambda}$ is updated every 200 sub-iterations (mini-batches).

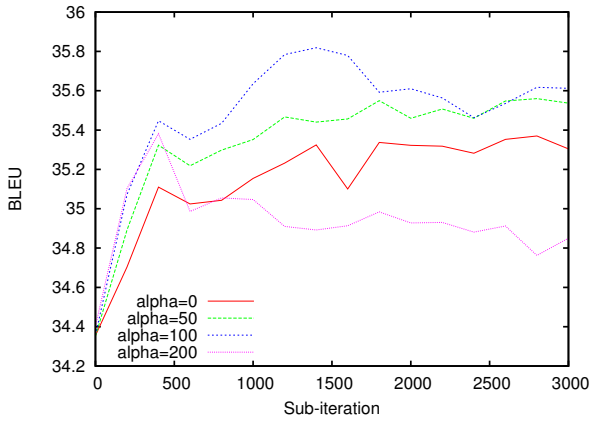


Figure 3: Evolution of BLEU scores on the dev set with different values of α. $\mathcal{L}_{pro-mm}$ is used in all cases.

4.3. Experimental results

The baseline, out-of-domain, system is used to generate the 300-best list for the in-domain corpus. It takes approximatively an half an hour if this process is parallelized by dividing the corpus in about 50 parts of $20,000$ sentences. δ is set to 250 (equations (6) and (7)) in all our experiments with the pairwise ranking criterion.

As reflected in equation (2), 4 translation models can be defined by various factorizations of $P(\mathbf{s}, \mathbf{t})$. For the sake of clarity, we focus our study on models estimating $P(\bar{t}_i|\bar{s}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1})$ and $P(\bar{t}_i|\bar{s}_{i-n+1}^{i-1}, \bar{t}_{i-n+1}^{i-1})$. We first compare the different objective functions defined in Section 3 and examine the impact of the margin on the former model. We then choose the best configuration to adapt the latter. Similar trends were observed with other CSTMs.

Figure 2 compares the three discriminative criteria respectively defined by (5), (6) and (8) in terms of BLEU scores on the dev set when adapting $P(\bar{t}_i|\bar{s}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1})$.

System	*dev*	*test*
Baseline systems (out-of-domain)		
n-code	33.9	27.6
n-code + CSTM WMT	34.4	28.5
Adapted systems		
n-code + CSTM CLL adapted	35.0	29.1
n-code + CSTM $\mathcal{L}_{mm}$ adapted $\alpha = 100$	35.1	29.4
n-code + CSTM $\mathcal{L}_{pro}$ adapted	35.4	29.5
n-code + CSTM $\mathcal{L}_{pro-mm}$ adapted $\alpha = 100$	35.8	29.6

Table 1: BLEU scores obtained for different adaptation schemes of the CSTM for $P(\bar{t}_i|\bar{s}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1})$ with WMT baselines: maximum conditional likelihood (CLL) *vs* discriminative adaptation. Log-linear coefficients of the baseline systems are re-tuned using the in-domain dev set.

Table 1 gives BLEU scores on both dev and test sets. According to these results, the pairwise ranking criterion, with or without max-margin((6) and (8)) clearly outperforms the max-margin approach (5) on the dev set. Further analyses (not detailed here) on each criterion's behaviour on the training set suggest that continuous space models quickly overfit the training data when adapted with the max-margin criterion. This result may outline the benefit of using criteria based on multiples hypotheses from different parts of the N-best list, rather than only on the best hypothesis and the most critical one as does the max-margin loss. Because of the superiority of the pairwise ranking approach, the rest of this section focuses on this criterion.

To assess the impact of the margin in $\mathcal{L}_{pro-mm}$, we plot on Figure 3 the evolution of the BLEU score on the dev set as a function of α. When $\alpha = 0$, the objective function only considers the pairwise ranking criterion $\mathcal{L}_{pro}$. By increasing α, we observe an improvement of 0.4 BLEU point, while beyond $\alpha = 100$, the performance starts to drop.

The results of adapting $P(\bar{t}_i|\bar{s}_{i-n+1}^{i}, \bar{t}_{i-n+1}^{i-1})$ are in Table 1. The upper part reports the baseline BLEU scores. Initial results were obtained with the out-of-domain one-pass system, and a 0.9 BLEU point improvement was obtained when reranking its output with the out-of-domain CSTM. The lower part of Table 1 summarizes the results obtained with various adaptation methods: the conditional likelihood (CLL) adaptation technique yields an additional increase of 0.6 BLEU point, which is nearly doubled when using the discriminative objective function $\mathcal{L}_{pro-mm}$ to perform adaptation. As showed in the middle part of Table 2, similar improvements are obtained with the adaptation of $P(\bar{t}_i|\bar{s}_{i-n+1}^{i-1}, \bar{t}_{i-n+1}^{i-1})$.

Finally, the lower part of Table 2 compares the performance obtained by our discriminative adaptation method to the one published in [9] for the same experimental setup. In our experiment (the last line), in-domain data are only used in two phases: the retuning of feature weights $\boldsymbol{\lambda}$; and the separate discriminative adaptation of two CSTMs. In [9], the

System	*dev*	*test*
Baseline systems (out-of-domain)		
n-code	33.9	27.6
n-code + CSTM WMT	34.6	28.2
Adapted systems		
n-code + CSTM CLL adapted	35.1	28.7
n-code + CSTM $\mathcal{L}_{pro-mm}$ adapted $\alpha = 100$	35.3	29.4
Model combination		
n-code (+TED) + all CSTMs CLL adapted [9]	36	29.7
n-code + all WMT CSTMs + 2 CSTMs $\mathcal{L}_{pro-mm}$	**36.4**	**29.9**

Table 2: BLEU scores obtained for different adaptation schemes of the CSTM for $P(\overline{t}_i|\overline{s}_{i-n+1}^{i-1}, \overline{t}_{i-n+1}^{i-1})$ in the middle part, and with model combination in the lower part. The notation n-code (+TED) emphasizes that for this system the baseline SMT system is *re-trained* with out-of-domain and in-domain data, while in all other cases the baseline system only uses out-of-domain data.

SMT system is entirely re-trained from scratch to integrate in-domain data (from word alignments to the large scale target language model), and all four CSTMs defined by (2) are adapted using the CLL criterion. This experiment shows that we can achieve slightly better performance by only adapting two CSTMs with the proposed objective function.

5. Related work

Most recent works in domain adaptation for SMT focuses on the modification of the sufficient statistics required by conventional discrete models [3, 4, 38], or on data selection [5, 6]. Our work owes much to recent contributions in discriminative training and tuning of SMT systems. While perceptron-based learning has been first introduced in [39, 40], margin-based algorithms such as MIRA [20, 21] are nowadays considered as more efficient to train Feature-Rich Translation systems. This property is especially relevant in our case, since we intend to learn a large set of parameters ($\boldsymbol{\theta}$). Another trend considers the optimization problem as ranking [41, 39, 22, 23]. Note that the ranking task corresponds to the integration of the CSTM that is actually used for N-best reranking. In this work, the proposed objective functions borrow from these two lines of research to both adapt the CSTM ($\boldsymbol{\theta}$) and tune its contribution ($\boldsymbol{\lambda}$) to the whole SMT system.

To the best of our knowledge, the most similar work on discriminative training or adaptation of neural network models is [12]. In this article, the authors propose to estimate the parameters of a neural network towards the expected BLEU score, while tuning $\boldsymbol{\lambda}$ by standard tools. Algorithm 1 is very similar to the optimization algorithm they describe, except that in our case, the feature weights $\boldsymbol{\lambda}$ are regularly updated for a better and tighter integration of the CSTM into the SMT system. Moreover, their proposed model only considers phrase pairs in isolation, while we use a probabilistic model of the joint distribution of sentence pairs. Expected BLEU training was also applied to recurrent neural network language model in [42].

In [13], the authors also introduce a ranking-type objective function that only aims to estimate word embeddings in a multitask-learning framework. Furthermore, [17] investigates the use of a large-margin criterion to train a recursive neural network for syntactic parsing. Interestingly, their model is also used to rerank N-best derivations generated by a conventional probabilistic context-free grammar. However, as showed by experimental results, the max-margin criterion alone is less adapted to machine translation. One explanation is that the N-best lists generated by the SMT system are not sufficiently diverse.

6. Conclusions

This paper has proposed and evaluated the use of discriminative criteria to adapt continuous space translation models. Instead of using a standard maximum likelihood method, the newly proposed algorithm discriminatively contrasts good and bad hypotheses from an N-best list produced by the baseline system into which the CSTM will be incorporated. A new adaptation method has been tested, consisting in jointly optimizing parameters from the neural network and from the SMT system so that the algorithm directly improves the system's overall quality. BLEU-based margins have also been included into these new loss functions and are proved to be useful. Our experiments consist in adapting out-of-domain CSTMs using a small quantity of in-domain parallel data, while keeping intact the out-of-domain baseline system. Our conclusions are two-fold. Firstly, we prove empirically the effectiveness of using discriminative criteria to adapt CSTMs, compared to the traditional maximum likelihood method. Secondly, our comparison shows that the pairwise ranking criterion is more suitable to Discriminative Reranking task in SMT than the max-margin approach, and that combining both criterion can deliver additional gains. In general, this work confirms the effective use of neural networks in Domain Adaptation for SMT systems.

For future work, we plan to combine our framework with other objective functions on N-best lists, such as *expected* BLEU [43]. We will also try an intensified use of the proposed algorithm by iteratively adding multiple feature functions into the SMT system; each model is trained using baseline system's N-best lists rescored with previously added models, in the hope that each model will capture complementary information and correct errors of the previous pass. Moreover, even though this work focuses on probabilistic n-gram translation models, our framework could be applied to any model structure [44, 18, 11] giving a score to each translation hypothesis.

7. References

[1] H. Daume III and D. Marcu, "Domain adaptation for statistical classifiers," *Journal of Artificial Intelligence Research*, vol. 26, pp. 101–126, 2006.

[2] J. Blitzer, "Domain adaptation of natural language processing systems," Ph.D. dissertation, University of Pennsylvania, 2008.

[3] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," in *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, 2007, pp. 128–135.

[4] N. Bertoldi and M. Federico, "Domain adaptation for statistical machine translation with monolingual resources," in *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, 2009, pp. 182–189.

[5] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011, pp. 355–362.

[6] R. Sennrich, "Perplexity minimization for translation model domain adaptation in statistical machine translation," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 539–549.

[7] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[8] H. Schwenk, M. R. Costa-jussa, and J. A. R. Fonollosa, "Smooth bilingual n-gram translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Prague, Czech Republic, 2007, pp. 430–438.

[9] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous space translation models with neural networks," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Montréal, Canada, 2012, pp. 39–48.

[10] Y. Hu, M. Auli, Q. Gao, and J. Gao, "Minimum translation modeling with recurrent neural networks," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 20–29.

[11] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[12] J. Gao, X. He, W.-t. Yih, and L. Deng, "Learning continuous phrase representations for translation modeling," in *Proc. ACL*, 2014.

[13] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[14] H. Schwenk, "Continuous space language models," *Computer Speech and Language*, vol. 21, no. 3, pp. 492–518, July 2007.

[15] T. Mikolov, S. Kombrink, L. Burget, J. Cernocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proceedings of ICASSP*, 2011, pp. 5528–5531.

[16] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language models for speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 1, pp. 197–206, 2013.

[17] R. Socher, J. Bauer, C. D. Manning, and N. Andrew Y., "Parsing with compositional vector grammars," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Sofia, Bulgaria, 2013, pp. 455–465.

[18] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, Washington, USA, 2013, pp. 1700–1709.

[19] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland, 2014, pp. 1370–1380.

[20] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki, "Online large-margin training for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Citeseer, 2007.

[21] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, June 2012, pp. 427–436.

[22] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., July 2011, pp. 1352–1362.

[23] P. Simianer, S. Riezler, and C. Dyer, "Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2012, pp. 11–21.

[24] M. Federico, S. Stüker, L. Bentivogli, M. Paul, M. Cettolo, T. Herrmann, J. Niehues, and G. Moretti, "The IWSLT 2011 evaluation campaign on automatic talk translation," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), 2012.

[25] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. Fonollosa, and M. R. Costa-Jussà, "N-gram-based machine translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[26] J. M. Crego and J. B. Mariño, "Improving statistical MT by coupling reordering and decoding," *Machine Translation*, vol. 20, no. 3, pp. 199–215, 2006.

[27] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language model," in *Proceedings of ICASSP*, 2011, pp. 5524–5527.

[28] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21, 2008, pp. 1081–1088.

[29] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[30] J. M. Crego, F. Yvon, and J. B. Mariño, "N-code: an open-source bilingual N-gram SMT toolkit," *Prague Bulletin of Mathematical Linguistics*, vol. 96, pp. 49–58, 2011.

[31] J. Niehues and A. Waibel, "Continuous space language models using restricted Boltzmann machines." in *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*, Hong-Kong, China, 2012, pp. 164–170.

[32] A. Vaswani, Y. Zhao, V. Fossum, and D. Chiang, "Decoding with large-scale neural language models improves translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, 2013, pp. 1387–1392.

[33] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Machine learning*, vol. 37, no. 3, pp. 277–296, 1999.

[34] R. McDonald, K. Crammer, and F. Pereira, "Online large-margin training of dependency parsers," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005, pp. 91–98.

[35] M. Collins, "Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002, pp. 1–8.

[36] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *in Proceedings of the 40th Annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[37] F. Casacuberta and E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30, no. 3, pp. 205–225, 2004.

[38] B. Chen, R. Kuhn, and G. Foster, "Vector space model for adaptation in statistical machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013, pp. 1285–1293.

[39] L. Shen and A. K. Joshi, "Ranking and reranking with perceptron," *Machine Learning*, vol. 60, no. 1-3, pp. 73–96, 2005.

[40] P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar, "An end-to-end discriminative approach to machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2006, pp. 761–768.

[41] L. Shen, A. Sarkar, and F. J. Och, "Discriminative reranking for machine translation." in *HLT-NAACL*, 2004, pp. 177–184.

[42] M. Auli and J. Gao, "Decoder integration and expected bleu training for recurrent neural network language models," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, June 2014, pp. 136–142.

[43] J. Gao and X. He, "Training mrf-based phrase translation models using gradient ascent," in *Proceedings of NAACL-HLT*, 2013, pp. 450–459.

[44] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks." in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013, pp. 1044–1054.

Extracting Translation Pairs from Social Network Content

Matthias Eck, Yury Zemlyanskiy, Joy Zhang, Alex Waibel

Facebook, Inc.
eck@fb.com, urikz@fb.com, joyzhang@fb.com, waibel@fb.com

Abstract

We introduce two methods to collect additional training data for statistical machine translation systems from public social network content. The first method identifies multilingual content where the author self-translated their own post to reach additional friends, fans or customers. Once identified, we can split the post in the language segments and extract translation pairs from this content. The second methods considers web links (URLs) that users add as part of their post to point the reader to a video, article or website. If the same URL is shared from different language users, there is a chance they might give the same comment in their respective language. We use a support vector machine (SVM) as a classifier to identify true translations from all candidate pairs. We collected additional translation pairs using both methods for the language pairs Spanish-English and Portuguese-English. Testing the collected data as additional training data for statistical machine translations on in-domain test sets resulted in very significant improvements of up to 5 BLEU.

1. Introduction

Current social networking websites like Facebook, Twitter and LinkedIn are operating globally. The majority of Facebook's over 1 billion users[1] are located outside of the US and user generated content is produced in a wide variety of languages. A globalized world also supports friendships across country and language barriers and makes news and entertainment sources in other languages easily accessible. It is Facebook's stated mission to make the world more open and connected and giving people the power to share.
All of these facts generate the need for translation of user content. Efficiency and especially the amount of content requested to translate make only automatic translation systems feasible.

One of the main challenges in training translation systems for social media content is the lack of in-domain training data. Bilingual corpora are generally only available in news or parliament domains, which are considerably different from the actual content that needs to be translated in social media applications.
Social media content frequently exhibits slang terms, colloquial expressions and other features not common in carefully edited news sources. Spelling errors are also very frequent. Social media content in Spanish and Portuguese specifically often exhibits a lack of correct diacritical marks.

A general approach to overcome any domain-mismatch problem is to somehow collect additional in-domain training data to augment the out-of-domain training data. Many experiments could show that this often significantly improves the translation performance.

The source that is used here is the actual social network. This paper introduces two different approaches to automatically collect parallel training data from social network content.

1.1. Multilingual Posts

Posting the same content in many languages is an approach that many fan pages, but also individual persons take to reach different groups of their friends and fan bases. Popular fan pages on Facebook have up to 100 million and more fans. As of August 2014 e.g. singer Shakira has 102 million fans, soccer club FC Barcelona has 72 million fans and soccer player Lionel Messi has 69 million. All three are examples of fan pages that post most of their updates in English and Spanish (also Catalan in FC Barcelona's case). Figure 1 shows an example post by Lionel Messi.

Figure 1: Multilingual post by Lionel Messi in Spanish and English

> Gracias a mis compañeros por elegirme como uno de los capitanes del equipo y por la confianza que han depositado en mí. Un abrazo.
>
> Thanks to my teammates for picking me as one of the club captains and for the confidence they have given me. A hug.

These are just some of the millions of pages on Facebook. It is likely that many of them have a multilingual group of people following the page. In order to serve these people better a large number have resorted to multilingual posts. This is even the case for pages of smaller, local businesses. Many cities and communities in the United States for example have large ethnic minority populations, most notably people of Hispanic and Asian descent. To reach these potential

[1] Facebook has 1.35B monthly active users as of Sept. 30th, 2014 (Q3 2014 earnings call)

customers even small businesses often resolve to multilingual communication. These pages and users want to ensure that all language groups of their fans are appropriately informed without relying on machine translation, which might not be available on all platforms.
Our first approach determines if an individual post is part of this category and contains more than one language. Should this be the case the post is split into the individual language segments and a classifier decides if the parts are indeed translations of each other.

1.2. URL Sharing

The second approach exploits the sharing function in Facebook allowing users to publicly share and re-share links to videos or other websites. Users on Facebook and other social networks use this function to point their friends and colleagues to interesting content and can also comment on it separately. Popular videos, articles and websites are shared many times even across different language users.
The assumption here is that two users or pages talking about the same content might have very similar comments. Therefore we can consider the respective posts *comparable* and we try to find true parallel sentences among them. It is for example rather common for users to translate movie titles or to quote important parts of a news article in their own languages.
Recently, the official "The Beatles" page shared a YouTube video featuring Paul McCartney and wrote a description about it. The hotel "Bayres Bohemios" in Argentina then decided to share the video with its guests. They posted the same link with the same description translated to Spanish (see Figure 2)

Figure 2: Descriptions by the pages "The Beatles" and "Bayres Bohemios" for the same URL

URL:
https://www.youtube.com/watch?v=pE_1V0phMW8

"The Beatles":
Paul is interviewed in this week's NME Magazine, which is on the stands from today.
In the article Paul discusses the recording process and working with the four producers who helped put together his 'New' album; Paul Epworth, Ethan Johns, Giles Martin and Mark Ronson. The article reveals the name of two of the tracks from the album; 'Alligator' and 'Save Us'.

"Bayres Bohemios":
Paul es entrevistado en la revista NME de esta semana, que está en las gradas de hoy.
En el artículo de Pablo discute el proceso de grabación y el trabajo con los cuatro productores que ayudaron a armar su disco 'New', Paul Epworth, Ethan Johns, Giles Martin y Mark Ronson. El artículo revela el nombre de dos de las canciones del álbum, 'Alligator' y 'Save Us'.

The rest of the paper will discuss some related work in section 2 and describe our methods in sections 3 and 4. Sections 5 and 1 describe the data we were able to collect and our experimental results using this data to improve machine translation systems for Spanish-English and Portuguese-English.

2. Related work

Collecting corpora for machine translation is a well-researched problem. Collecting additional parallel sentences from Wikipedia and the web itself has been extensively studied due to the ease of access. [1]–[5]. Most approaches consist of two steps, identifying comparable candidate segment pairs based on some connection feature between them and a final step to classify the found candidate segments into actual translation pairs. A classification approach similar to [6] is generally applied. The importance of the accuracy of the classification is generally closely related on the method used to identify candidate segments.
Closely related to our multi-lingual post approach is the work done in [7] to collect additional Chinese-English translation pairs from Sina Weibo content. The authors continue the work in [8] by using crowdsourcing to improve the accuracy of the extracted data.

3. Collecting from multilingual Facebook posts

For all discussed experiments, only public posts were considered and in all instances these public posts were stripped of specific user attribution.

We generally consider all (public) Facebook posts as candidates for multilingual posts. At creation time of every Facebook post, a standard language identification system is applied. This helps with News Feed ranking and later the ability to show appropriate automatic translations.
Our translation extraction approach is now focusing on one source and target language pair at a time and we consider all posts that were identified as either target or source language in this step. The standard language identification does not consider multilingual posts and will only assign a single language identifier.

3.1. Language identification and segmentation

To identify the segments, we first apply an additional language identification step and decide for each unigram what its most likely language is.
Once the basic language identification is applied we also check if the ratio of terms identified as either language is within a reasonable range, otherwise the post is already discarded as unlikely to contain translated segments e.g. a post that contains ten English words and only one Spanish word.

In a second language identification step we apply a smoothing on the identified languages to eliminate spurious incorrect identifications. This changes the identified language of a single word if the neighboring words were identified as the other language. This has proven helpful for misspellings. Table 1 shows an example for a misspelling "mi" in the English segment. This is initially incorrectly identified as Spanish and then fixed in the smoothing step

Table 1: Language ID with smoothing

	Happy	birthday	mi	brother	…
Language ID	en	en	es	en	…
Smoothed	en	en	en	en	…

Once the language of every word has been identified the post is split into the two longest segments, which are then classified to determine if they are actually translations of each other.

3.2. Classifying the translation

All translation classifiers that were applied in this work are based on seed lexicons taken from the baseline trainings for each translation direction. This especially provides word-to-word lexicons to the classifiers.

Experiments have shown that in the multilingual post case even simple word-to-word translation heuristics provide adequate performance to distinguish candidates that are translations from ones that are not. The reason seems to be that in this case the users either actually provide a translation or they code-switched in their posts. In this case the segment contents are not close. An example post for this is "*quality time con mi chiqui"*[sic]. In this case there is little danger that the two segments could be classified as translations since no part of the segments are translations or even semantically close.

It is obviously also possible to apply more sophisticated segment classification and we describe a detailed model in section 4.2 originally developed to classify candidates generated from URL shares where candidates can often be much closer. The actual experiments reported all used the classifier described in section 4.2.

4. Collecting translations from URL Shares

An alternative idea to extract translations from Facebook posts is to try to find monolingual posts that are translations of each other. Of course it is not practical or reasonable to compare every post with every other post, so the idea is to preselect post pairs that are comparable, i.e. discuss the same content.
Our idea was to look at URL shares. Users in Facebook (and other social networks) have the ability to post links to web content outside of the social network. Should two users link to the same URL they are obviously commenting on the same content and it is likely that some of those users comments could be translations of each other.
Some examples are translated quotes from a news article, translated song, movie or book titles or just general comments like "*Great game by Germany in the world cup*". Given the vast number of users on popular social networks it is likely that a small number of them will then be actual translations that can be collected.

4.1. Collecting URL shares

As stated, the task of searching for parallel sentences in all possible combinations of monolingual posts is intractable. In addition to considering only monolingual posts in different languages, which shared the same URL, we also used a couple of other simple heuristics to further reduce the search space.
We split each post into individual sentences and compare all sentences in one language with sentences in other languages using these simple rules:

- Original posts share the same URL
- At most a length ratio of 2
- Difference between posts' creation times is no more than 3 days
- Three sequential words in one sentence translate with high lexical probability into three other sequential words in the other sentence.

These procedures can be efficiently performed in a MapReduce framework handling an enormous amount of data.
If we find a match between sentence A from post A^* and sentence B from post B^* we mark all possible pairs from A^* and B^* as candidates. This algorithm does not take the translation direction into account, so it has to be performed once per language pair.
Overall we identified 25 million candidate pairs for Portuguese-English and 9 million for Spanish-English (in the chosen timeframe).

4.2. Translation classifier

The final step is to filter parallel sentences from the prepared candidate pairs. It has been shown (in [9]–[11]) that SVM-based classifiers with lexical features are performing quite well for this purpose.
We rely on a combination of 25 features selected from [9]–[11]:

- ratio of number of words per sentence
- all-to-all alignment features (per each direction)
  - total IBM score (with all-to-all alignment)
  - maximum fertility
  - number of covered words
  - length of longest sequence of covered words
  - length of longest sequence of not-covered words;

Also all features except the IBM score are normalized by source sentence length.

- max alignment (per each direction)
  - total IBM score
  - top 3 fertility values for target sentence
  - number of covered words for target sentence
  - "maximum intersection": maximal number of consequent source words, which have corresponding consequent target words
  - maximum number of consequent uncovered words in target sentence

Here all features are normalized by target sentence length except the IBM score (which is not normalized) and maximal intersection (which is normalized by source sentence length).

We used the same parallel corpora from the baseline machine translation training and tuned the classifier in order to achieve 95%-98% precision on the dataset. A possible problem here is that the data and users posts are essentially in different domains and the classifier might perform worse on our candidate pairs. It is common practice in this case ([11]) to run the filtering iteratively – using updated lexical dictionaries every time. However, it appeared to not be required, as the extracted corpora from the first iteration already gave a significant boost in translation quality.

The results show that the classifier filtered out 99% of the candidate pairs, but the remaining 1% was of very good quality – we did not find any non-parallel sentences while inspecting. The most common error was a few extra words in one of the sentences. The results show, that this does not negatively affect the final performance. Word and phrase extraction is generally robust if this does not occur too frequently.

5. Data Collection Statistics

Data for both methods was collected from public Facebook posts. The collected data is not directional and we used the data sets for tests in both directions. Table 2 shows the exact statistics for the collected data.

Table 2: Collected data statistics

	Es-En	Pt-En
Baseline data	*500,000 lines* 8.48M/8.44M Es to En 9.29M/10.06M En to Es	*500,000 lines* 11.29M/11.26M Pt to En 11.26M/12.24M En to Pt
Multilingual posts	17,214 lines 925k Es words 925k En words	6,208 lines 241k Pt words 236k En words
URL shares	120,594 lines 2.91M Es words 2.73M En words	95,444 lines 2.35M Pt words 2.28M En words

Spanish is more common on Facebook than Portuguese, which explains why more data could be collected for Spanish-English compared to Portuguese-English.

6. Translation Experiments

The developed methods were tested on two language pairs, Spanish-English and Portuguese-English for both translation directions each.

6.1. Training and Testing Data

For both language pairs development and test sets were created from manually translated public Facebook posts. Approximately 2,000 lines were translated and split into development and test sets.

The selected posts had previously been requested for automatic translation for the respective language pair, so they are exactly in-domain for the task and exhibit all the typical features.

The training data consists of out-of-domain data taken from European Parliament data (EPPS) and general phrases from the Tatoeba corpus[1]. The training data was sorted according to estimated importance [12] and only the top 500k sentence pairs were included in the training. The results showed that this did not result in any significant drop in translation performance and allowed for much faster training runs.

6.2. Machine Translation System

We used the open-source Moses statistical machine translation system [13]. All systems were trained following the standard training method using the parallelized implementation mgiza of giza++ [14], [15] and standard phrase extraction. The language models were regular 3-gram models with Kneser-Ney discounting. They were trained on the target side of the training data using the SRI toolkit [16], [17]. We applied standard minimum error rate training on our development sets and tested the systems on the separate test sets. All systems were evaluated using the standard BLEU metric [18].

6.3. Experimental Results

The experimental results in Table 3 illustrate the improvements for all four translation directions. Starting from the baseline scores we see varying improvements of up to 5.2 BLEU when using either approach. Even though the URL shares collected significantly more data, the multilingual post approach also results in significant BLEU improvements and it outperforms the approach for Spanish to English.

Combining both data sources generally further improves the performance, which indicates that the data collected is considerably different from each other. Inspection of the data confirmed this and it appears that the data from multilingual posts often contains sales offers and local events while the data collected from URL shares covers more popular culture, entertainment and politics.

[1] http://tatoeba.org

We also calculated the (token) out-of-vocabulary (OOV) rates for each dataset and this further explains the improvements. In every case the added data significantly improves the OOV situation. This is due to improved coverage of spelling errors, slang terms and Internet lingo.

The results also show that the URL shares approach generally gives greater improvements than the multilingual post extraction (with the exception of Spanish to English). The data extracted from multilingual posts does especially not perform very well for translations from English to Spanish or Portuguese, while it performs better for translations into English.

Table 3: Experimental Results – BLEU (token OOV rate in parentheses)

	Es→En	En→Es
Baseline	22.08 (8.7%)	22.48 (12.9%)
+multi	23.47 (7.8%)	22.72 (12.0%)
+shares	23.16 (6.0%)	27.61 (10.4%)
+multi+shares	24.30 (5.9%)	27.78 (10.2%)
	Pt→En	En→Pt
Baseline	28.39 (7.9%)	26.87 (10.8%)
+multi	28.92 (7.6%)	26.95 (10.5%)
+shares	31.34 (6.9%)	31.11 (9.1%)
+multi+shares	31.67 (6.8%)	30.92 (9.0%)

6.4. Example translations

In addition to the standard automatic BLEU metric we also analyzed how the additional data actually improved our translation systems by comparing baseline and improved translations. Table 4 shows some example translations from the Spanish to English translation system with the source and reference translations.

The first translation is a typical example of a concept "memory card" that is unlikely to be present in the out-of-domain data.

The second example illustrates an out-of-vocabulary term "agrego", which is not present in the baseline system and is then covered in the improved system. It also shows that the term "like" is directly used in Spanish instead of a Spanish term.

The next example shows how the translation of the Spanish term "cumple" is changed from the incorrect "meets" and the last example again contains a regular OOV term "cargador" that is not covered previously.

Table 4: Experimental Results - Example translations

Source	sin tarjeta de memoria .
Baseline	without card by heart
Improved	without memory card
Reference	without memory card
Source	like y agrego !!
Baseline	like and *agrego*!!
Improved	like and add!!
Reference	like and add!!
Source	feliz cumple preciosa !
Baseline	happy meets beautiful
Improved	happy birthday beautiful!
Reference	happy birthday, honey!
Source	con el cargador incluido.
Baseline	with the *cargador* included.
Improved	with the charger included.
Reference	charger included.

7. Conclusion

We presented two methods to collect additional translation pairs from public social network content, specifically public Facebook posts. First, we identified multilingual posts, where the actual posts contain their own translation. We also investigate extraction from "comparable" public posts identified by sharing the same URL.

Using both methods we are able to collect significant additional bilingual training data for the language pairs Spanish-English and Portuguese-English. Adding the collected data from either method to the overall training data improves the translation performance significantly with overall improvements of up to 5.2 BLEU. The main improvements are caused by enhanced vocabulary and phrase coverage of social network content. Both methods appear to collect data in slightly different topics and style, so the improvements are complementary and add up to combined higher scores.

Collecting translations based on the URL shares approach has the additional advantage to not be limited by language pairs that have a lot of need for multilingual posts and bilingual speakers; instead it can be more generally applied to any language pair.

8. References

[1] P. Resnik and N. A. Smith, "The Web as a Parallel Corpus," *Journal of Computational Linguistics*, vol. 29. pp. 349–380, 2003.

[2] Y. Zhang, K. Wu, J. Gao, and P. Vines, "Automatic Acquisition of Chinese-English Parallel Corpus from the Web," in *Proceedings of the 28th European Conference on Advances in Information Retrieval (ECIR 2006)*, 2006.

[3] K. Fukushima, K. Taura, and T. Chikayama, "A Fast and Accurate Method for Detecting English-Japanese Parallel Texts," in *Proceedings of the Workshop on Multilingual Language Resources and Interoperability (MLRI 2006)*, 2006.

[4] J. Uszkoreit, J. M. Ponte, A. C. Popat, and M. Dubiner, "Large Scale Parallel Document Mining for Machine Translation," in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, 2010.

[5] F. Ture and J. Lin, "Why Not Grab a Free Lunch?: Mining Large Corpora for Parallel Sentences to Improve Translation Modeling," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT 2012)*, 2012.

[6] D. S. Munteanu and D. Marcu, "Improving Machine Translation Performance by Exploiting Non-Parallel Corpora," *Journal of Computational Linguistics*, vol. 31, no. 4. MIT Press, Cambridge, MA, USA, pp. 477–504, Dec-2005.

[7] W. Ling, G. Xiang, C. Dyer, A. Black, and I. Trancoso, "Microblogs as Parallel Corpora," in *Proceedings of the 51st Annual Meeting on Association for Computational Linguistics (ACL 2013)*, 2013.

[8] W. Ling, L. Marujo, C. Dyer, A. Black, and I. Trancoso, "Crowdsourcing High-Quality Parallel Data Extraction from Twitter," in *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT 2014)*, 2014.

[9] T. Herrmann, M. Mediani, J. Niehues, and A. Waibel, "The Karlsruhe Institute of Technology Translation Systems for the WMT 2011," in *Proceedings of the Sixth Workshop on Statistical Machine Translation (WMT 2011)*, 2011.

[10] J. R. Smith, C. Quirk, and K. Toutanova, "Extracting parallel sentences from comparable corpora using document level alignment," in *Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT 2010)*, 2010.

[11] S. Hewavitharana, "Detecting Translational Equivalences in Comparable Corpora," 2012.

[12] M. Eck, S. Vogel, and A. Waibel, "Low Cost Portability for Statistical Machine Translation based on N-gram Frequency and TF-IDF," in *International Workshop on Spoken Language Translation (IWSLT 2005)*, 2005.

[13] P. Koehn, H. Hoang, and A. Birch, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, 2007.

[14] Q. Gao and S. Vogel, "Parallel Implementations of Word Alignment Tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, 2008, pp. 49–57.

[15] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Journal of Computational Linguistics*, vol. 29. pp. 19–51, 2003.

[16] A. Stolcke, "Srilm — an Extensible Language Modeling Toolkit," *Proceedings of the International Conference on Spoken Language Processing (ICSLP 2002)*, vol. 2. 2002.

[17] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "SRILM at Sixteen : Update and Outlook," in *Automatic Speech Recognition and Understanding Workshop (ASRU 2011)*, 2011.

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*, 2002.

An Exploration of Segmentation Strategies in Stream Decoding

Andrew Finch *Xiaolin Wang* *Eiichiro Sumita*

Multilingual Translation Group
National Institute of Information and
Communications Technology
Kyoto, Japan
{first.last}@nict.go.jp

Abstract

In this paper we explore segmentation strategies for the stream decoder - a method for decoding from a continuous stream of input tokens, rather than the traditional method of decoding from sentence segmented text. The behavior of the decoder is analyzed and modifications to the decoding algorithm are proposed to improve its performance. The experimental results show our proposed decoding strategies to be effective, and add support to the original findings that this approach is capable of approaching the performance of the underlying phrase-based machine translation decoder, at useful levels of latency. Our experiments evaluated the stream decoder on a broader set of language pairs than in previous work. We found most European language pairs were similar in character, and report results on English-Chinese and English-German pairs which are of interest due to the reordering required.

1. Introduction

Statistical machine translation (SMT) technology has advanced to the point where it is becoming capable enough to be useful for many applications. The process of automatic simultaneous interpretation however is another matter entirely. The interpretation process is difficult, even for skilled human interpreters, and presents a major challenge to a machine the since in addition to the translation process, decisions need to be made about when to commit to outputting a partial translation. Such decisions are critical since once such an output is made it can be difficult and highly undesirable to correct it later if it is in error.

In simultaneous interpretation, the input to the automatic interpretation system is often a continuous stream of tokens. Since the output from the system occurs periodically, the output of the system is segmented. In order to produce this output segmentation two strategies can be employed. In the first, the stream is segmented before the machine translation process begins, and the machine translation system is constrained to translate using the given segmentation. In order to distinguish the methods that segment the input prior to the decoding in a pre-processing step, we will refer to them as "pre-segmentation" in this paper. In the second, the segmentation process is performed during the decoding of the input stream. The work presented here is primarily concerned with the latter, but proposes and evaluates a method to integrate them.

2. Related Work

The work in this paper is based upon the *stream decoder* [1], an extension to a phrase-based statistical machine translation decoder that allows it to decode directly from continuous stream of tokens. We describe this methodology in more detail in Section 3.

In [2] the prosody information in the speech signal was used to segment a continuous stream of speech input for translation. In their experiments, a silence duration of approximately 100ms was found to be suitable for segmentation.

A number of diverse strategies for pre-segmentation were studied in [3]. They studied both non-linguistic techniques, that included fixed-length segments, and a "hold-output" method. The hold-output method method is relevant to the research in this paper because it relies the same principle used by the stream decoder. It identifies contiguous blocks of text that do not contain alignments to words outside them. An SVM was used to predict these blocks prior to the decoding process; the stream decoder operates by identifying similar structures during decoding. Their experimental results showed this method to be ineffective. Linguistically-motivated segmentation techniques were also considered. Conjunctions, sentence boundaries and commas were investigated, with commas being the most effective segmentation cue in their investigation.

In [4] a strategy for pre-segmentation based on searching for segmentation points while optimizing the BLEU score was presented. An attractive characteristic of this approach is that the granularity of the segmentation can be controlled

by choosing the number of segmentation boundaries to be inserted, prior to the segmentation process.

The automatic interpretation from English into Japanese has been studied in [5]. Their approach used heuristics to identify predicates that are likely to be invertible from a dependency structure derived from a phrase-structure parse of the English. They exploit the somewhat free word order of Japanese to re-order the Japanese tokens into an order that is appropriate for interpretation. The resulting word order may be a little dis-fluent, but is nonetheless grammatically valid and is typical of the kind of compromise that needs to be made during interpretation.

There are also some related studies in translation process research (for example, [6, 7]) that study in detail the process of human simultaneous interpretation.

In [8] it was shown that the prediction and use of soft boundaries in the source language text, when used as reordering constraints can improve the quality of a speech translation system.

3. Stream Decoding

The stream decoding strategy differs from approaches based on the pre-segmentation of the stream of input tokens in that the segmentation decisions are able to exploit information from the decoding process itself. In [9], it is stated that long segments of around 10-40 words are required in a pre-segmentation strategy in order to achieve performance close to the underlying machine translation system. These long segments give rise to long latencies, and the penalty for reducing the segment size in order to achieve acceptably latencies is typically severe. These issues have been addressed recently with more intelligent strategies for choosing the segmentation points [4], but nonetheless we believe the stream decoding approach deserves more attention in the literature, and merits further study for the following reasons:

- Stream decoding uses characteristics of the decoding process for segmentation, and requires no annotation of the input token stream.
- Stream decoding is able to enforce a maximum limit on the latency.
- The first results on English-Spanish translation ([1]) were very promising.

3.1. Overview of the Stream Decoding Process

The reader is referred to the original paper [1] for a complete description of the stream decoding process; in this section, for completeness, we provide a brief summary of the stream decoding methodology.

Figure 1 depicts a stream decoding process. The input to the stream decoder is a stream of tokens (it is also possible to configure the decoder to operate on tuples of confusable token sequences from a speech recognition decoder, but for the purposes of this paper we consider streams of tokens). A typical phrase-based machine translation system will decode token sequences, where a token (typically word) sequence usually represents a sentence in the source language. The decoder will construct a search graph from this sequence of tokens and output the n-best derivations of target token sequences from this graph.

The stream decoder, in contrast, operates on a potentially infinite sequence of tokens. As new tokens arrive, states in the search graph are extended with the new possible translation options arising from the new tokens. Periodically the stream decoder will commit to outputting a sequence of target tokens. At this point a state from the search graph is selected, the search graph leading from this state is kept, and the remainder discarded. The search then continues using the pruned search graph. In our implementation of the stream decoder the language model context is preserved at this state for use during the subsequence decoding. In this manner the stream decoder is able to operate on a stream of tokens that contains no segment boundary information. The segmentation occurs as a natural by-product of the decoding process.

3.2. Latency Parameters

The stream decoding process is governed by two parameters L_{max} and L_{min}. These parameters are illustrated in Figure 1. The L_{max} parameter controls the maximum latency of the system. That is, the maximum number of tokens the system is permitted to fall behind the current position. If interpreting from speech, the parameter represents the number of words the system is allowed to fall behind the speaker, before being required to provide an output translation. This parameter is a hard constraint that guarantees the system will always be within L_{max} tokens of the current last token in the stream of input tokens. The parameter L_{min} represents the minimum number of words the system will lag behind the last word spoken. It serves as a means of preventing the decoder from committing to a translation too early.

3.3. Determining the Segmentation Point

Algorithm 1 shows the algorithm used to select the segmentation point. The decoder maintains a sequence of tokens that represent the sequence of untranslated tokens from the input stream (see Figure 1). As new tokens arrive from the input stream, they are added to the end of the sequence. When the length of this sequence reaches L_{max}, the decoder is forced to provide an output. A search state is chosen from the sequence of states in the search graph representing the best hypothesis that covers the full sequence of untranslated words. In short, the best hypothesis is rolled back, state by state, until the remaining state sequence translates a contiguous sequence of source words starting from beginning of the sequence of untranslated words, and the number of words that would remain in the sequence of untranslated words after the translation is made, is at least L_{min}. It is possible that no

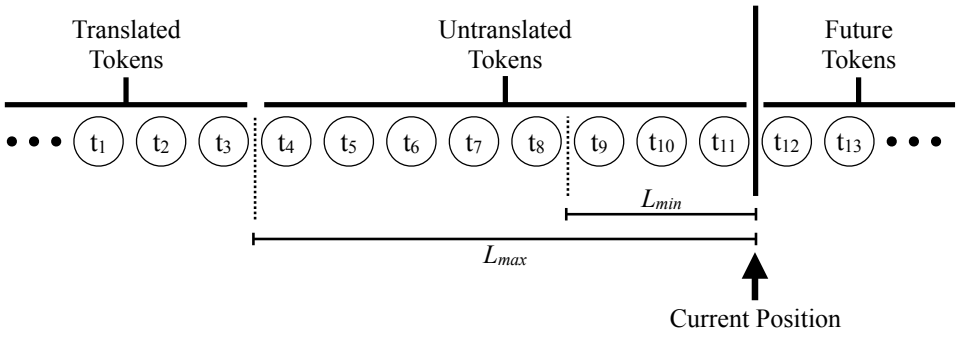


Figure 1: The stream decoding process.

Algorithm 1: Selecting a segmentation point.

Input: A sequence of search states $s_0, \ldots, s_n$ representing the best hypothesis. s_0 being the initial state, and s_n being the final state.
Output: A state $s_i \neq s_0$ representing the end of the translation segment, or s_0 if the process failed to find a suitable state.

```
foreach i = n to 1 do
    if the tokens translated by s_0 ... s_i are a
    contiguous sequence starting immediately after
    the last translated source token then
        if the number tokens translated by the states
        following s_i at least L_min then
            return s_i
        end
    end
end
return s_0
```

Language Pair	Training	Dev	Test
English-Spanish	180853	887	1701
English-Chinese	179651	887	1397
English-German	171721	887	1700

Table 1: Statistics on corpora using in the stream decoding experiments. The numbers given are in segments, representing individual subtitles, corresponding approximately to sentences.

such state exists, in which case the algorithm returns s_0, and since the stream decoder is required to make an output, it must use an alternative strategy.

In this alternative strategy, the stream decoder will undertake a new decoding pass in which it is forced to make a monotonic step as the first step in the decoding process. Then, a state is selected from the best hypothesis using Algorthim 1. This process may also fail if the monotonic step would lead to the violation of L_{min}. In our implementation, we allow the decoder to violate L_{min} only in this case.

4. Experimental Methodology

4.1. Corpora

For the experiments that explore the operation of and enhancements to the stream decoder we use the TED[1] talks data sets from the IWSLT2014 campaign. We studied English to: Spanish, Italian, French, German and Chinese, and found the results on the set of European language pairs were mostly similar in character, and we therefore report results on only English-Spanish (a typical pair) and English-German (an exceptional pair) from this set. Statistics on the corpora are given in Table 1. The European language data was tokenized by the Stanford PTBTokenizer. The Chinese was segmented using the Stanford Chinese word segmenter [10] according to the Chinese Penn Treebank standard.

4.2. Decoder

Our stream decoder was implemented within the framework of the OCTAVIAN decoder, a phrase-based statistical machine translation decoder that operates in a similar manner to the MOSES decoder [11]. The training procedure was quite typical: 5-gram language models were used, trained with modified Kneser-Ney smoothing; MERT [12] was used to train the log-linear weights of the models; the decoding was performed with no limit on the distortion.

4.3. Evaluation

The BLEU score [13] was used to evaluate the machine translation quality in all our experiments. Where sentence segmentation was known we used both talk and sentence-level BLEU, and for the experiments where true stream decoding was performed on a stream of tokens with no segmentation information, talk-level BLEU was used. In talk level BLEU each talk is considered to be a single sentence in the BLEU computation. For consistency only the talk level BLEU results are reported in this paper, but the results from the sentence-level BLEU experiments were similar in character.

[1] http://www.ted.com

5. Alternative Stream Decoding Strategies

5.1. Increasing the Output Frequency

5.1.1. Methodology

As explained in the previous section, in the originally proposed stream decoder, the best hypothesis is unrolled backwards until a suitable point is found for output. The principle here is to find the longest subsequence of states in the best hypothesis that satisfies the constraints that determine whether the segmentation point is permissible. However, other strategies are possible. One plausible strategy is instead of committing to the longest permissible output, commit to the shortest. The algorithm is identical to that shown of Algorithm 1 except that the "foreach" loop that ranges from $i = n \ldots 1$, ranges from $i = 1 \ldots n$. The approach takes less of a risk, since it will commit to shorter translations. On the downside, it will lag behind the original strategy given the same values for its latency parameters.

For this reason, it is unfair to compare these approaches under the constraint that their L_{max} and L_{min} parameters are the same, since there may be a bias in favor of the strategy that commits to the shortest output, and this strategy will gain its advantage by increasing the latency of the tokens in the output stream. To remove this potential bias, we therefore compare these two methods in the experiment below by measuring the trade-off between machine translation quality (measured using the BLEU score) and average latency per token L_{avg}. That is the average number of tokens each token is behind the input stream, given by:

$$L_{avg} = \frac{\sum_{i=1,N} L(i)}{N} \quad (1)$$

where N is the total number of words in the input stream, and $L(i)$ is the latency after word i has been processed.

5.1.2. Experiment

Figure 2 shows the results on English-to-Spanish translation task. Experiments were run for values of L_{max} in the range 1 to 10, and the points are annotated with these values. The oracle values of L_{min}, that is the value of L_{min} that gave rise to the highest BLEU score, were used. The graph plots L_{avg} against BLEU score for each experiment. For high and low values of L_{max} the two strategies are similar in performance, but for $3 \leq L_{max} \leq 6$ the strategy that makes more frequent, but shorter output is the better strategy. Of course there may be human factors to consider, but in terms of the machine translation evaluation scores at least, the shorter output strategy would seem to be the more effective approach, especially when lower latencies are required. This approach varied in its effectiveness across language pairs however, with some European language pairs (for example English-to-French) showing almost no difference in performance. The proposed strategy was always at least as good as the baseline, and therefore it was adopted in the remainder of the experiments reported here.

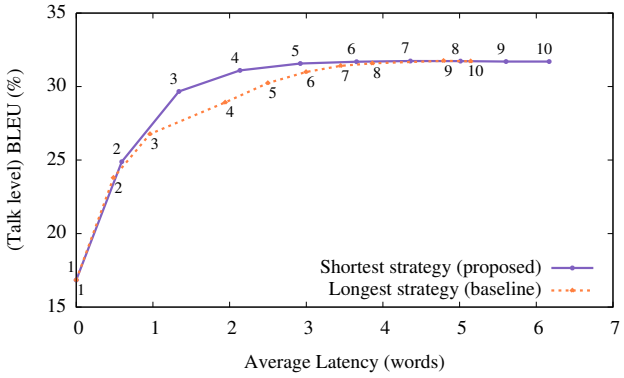


Figure 2: The trade-off between BLEU score and average latency for two different strategies for selecting the segmentation point.

5.2. Minimizing the Number of Forced Monotonic Steps

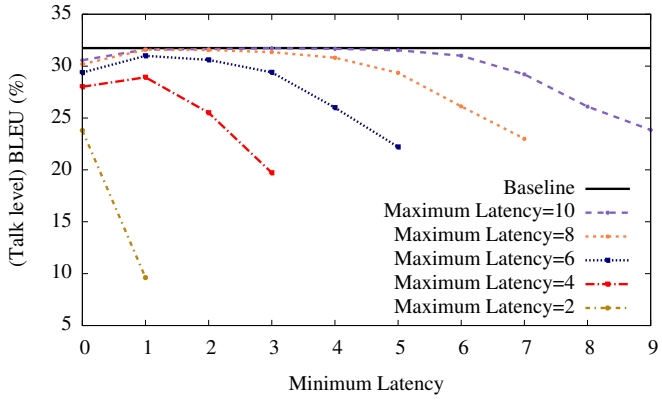
5.2.1. Motivation

In Section 3.3 it was explained that during stream decoding the best hypothesis is rolled back until a satisfactory segmentation point is found. In some cases, no such segmentation point exists and the decoder resorts to an alternative decoding strategy that forces the first step of the decoding process to be monotonic. This section is motivated by the concern that constraining the decoder in this manner will lead to translation hypotheses that diverge from the optimal path, impacting the overall translation performance. We therefore seek a method that can reduce the number of forced monotonic steps.

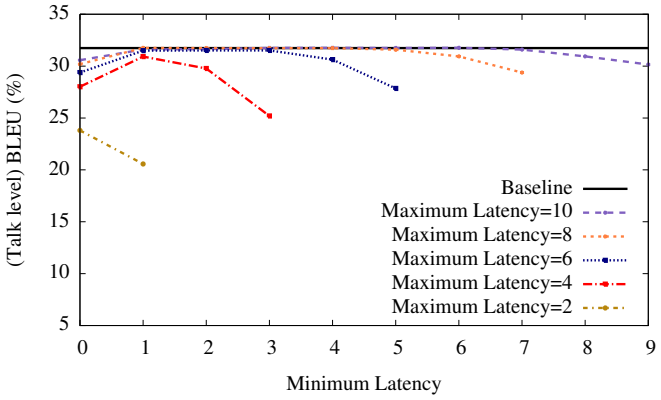
5.2.2. Methodology

One plausible method to alleviate the issue is to extend the stream decoding approach to allow it to select a segmentation point from the whole search graph, rather than from the 1-best hypothesis. We proposed a straightforward extension of the existing approach: to select a state from the n-best list. The proposed method applies Algorithm 1 iteratively over an n-best list of derivations, from rank 1 to n, terminating on the first rank in which a suitable segmentation point is found. Only if no segmentation point is found in the n-best list, does the decoder resort to a forced monotonic decoding step.

We analyzed the effect of the approach on the number of forced monotonic steps for English-Spanish. The results are shown in Figure 3, the oracle value of L_{min} is used. The figure shows the percentage of translated segments that were the result of a decoding hypothesis that contained a forced monotonic step. The results clearly show that the proposed method can have a substantial impact on the number of forced monotonic steps. We investigate whether or not this leads to an improvement in machine translation performance in the next sections.

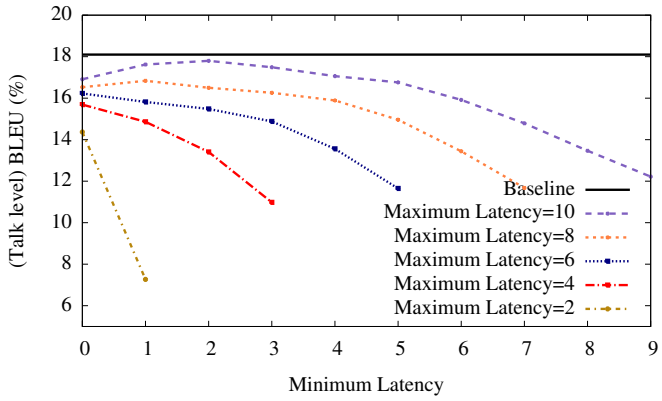


(a) Selecting from the best hypothesis.

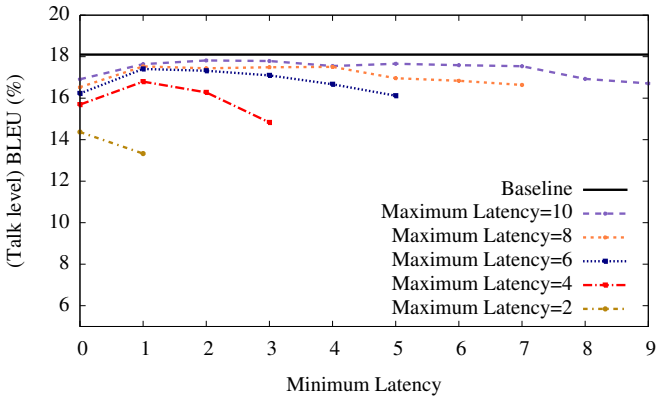


(b) Selecting from the 20-best hypotheses.

Figure 4: Using the n-best hypotheses to select the segmentation point for English-Spanish.



(a) Selecting from the best hypothesis.



(b) Selecting from the 20-best hypotheses.

Figure 5: Using the n-best hypotheses to select the segmentation point for English-Chinese.

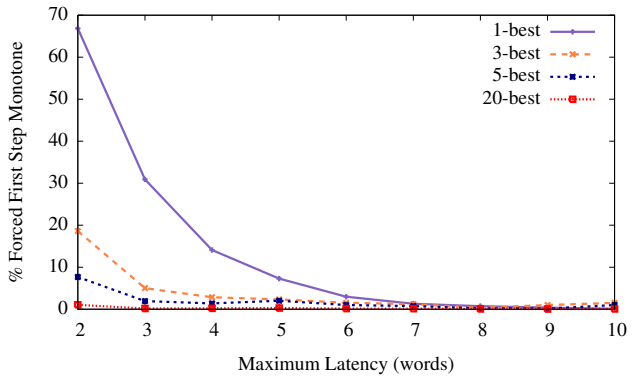


Figure 3: The proportion of output segments containing forced monotonic decoding steps for different length n-best lists.

5.2.3. *English-Spanish Translation*

Even though the number of forced monotonic decoding steps can be reduced by using an n-best list, it does not guarantee an improvement in performance. Selecting a state from a hypothesis other than the 1-best comes with a price as hypotheses further down the n-best list are likely to represent translations of lower quality.

Figure 4 shows the results of an experiment using the proposed method in the previous section on the English-to-Spanish task. The experiments used identical settings apart from the length of the n-best list used to select the segmentation point. The baseline on both graphs represents the performance of the underlying phrase-based SMT decoder when decoding the data according to the segmentation provided in the corpus.

The results show that the stream decoder, which must provide its own segmentation is able to achieve evaluation performance comparable to the baseline SMT system. The stream decoder may have been helped by the fact that the baseline system was decoding without a distortion limit. Typically languages such as English and Spanish, having similar word orders benefit from a constraint on the reordering, which the stream decoder may be providing as a consequence of more monotonic decoding process. Nonetheless

we feel its performance is impressive.

The results of this experiment show our proposed method is very effective in improving the stream decoder. There are two important differences in the graphs, firstly the curves do not drop as sharply as L_{min} is increased, making the approach less sensitive to the selection of this parameter. Secondly, and more importantly, the performance on the experiments with lower latencies (where L_{max} is less than 6), is improved overall. We ran a set of experiments on the English-Spanish task to determine the effect of varying the size of the n-best list. We found that the approach was not very sensitive to the size of the n-best list for small values of n. The best results were obtained with $5 \leq n \leq 20$.

5.2.4. Other Language Pairs

The original stream decoder was evaluated on an English-Spanish task, and for consistency with the original work, so far we have shown results on the same language pair (but a different corpus). We ran experiments on all of the languages for which data was provided for the IWSLT2014 machine translation shared tasks. The stream decoder proved robust to differences in the language pair chosen. The results were generally similar in character to those presented for English-Spanish. We have omitted these results for brevity, and instead present results on the English-Chinese and English-German pairs which are interesting because their word orders are not similar, and as a consequence a substantial amount of reordering is necessary in the decoding process. These language pairs were expected to present more of a challenge to the stream decoder.

We conducted the same experiment presented in the previous section on an English-to-Chinese task, and the results are shown in Figure 5. As expected, it can be seen in the Figure 5a that the cost in terms of BLEU score is greater when lower latencies are required than for English-to-Spanish. The results have the same general character as before; the use of the n-best list has improved the performance of the lower latency curves, and also made the decoder far less sensitive to variations in the L_{min} parameter.

Among the European languages, German has some significant structural differences that can be expected to create difficulties for simultaneous interpretation. We show the results on the English-German pair in Figure 6. The results appear similar to the English-Chinese results, with a larger penalty in BLEU for shorter latencies. Moreover, the curves on the graph fall more sharply than the other languages tested with increasing L_{min}, indicating that the stream decoder is more sensitive to the value chosen for this parameter.

5.3. Selecting the Most Productive State

Instead of selecting a state in the set of 1-best or n-best hypotheses according to the algorithms described in the previous section, it is also possible to use other criteria to select the search state from the full search graph. One plausible

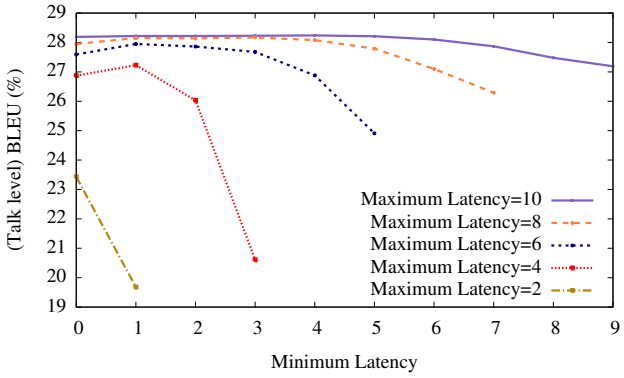


Figure 6: Performance on the English-German task (Using a 20-best list).

heuristic is to select the state that is in the greatest number of search paths leading to the final stack; the "most productive" state. The intuition behind this idea was that this state might provide the greatest number of good alternative search paths for decoding the future tokens. In the event of a tie in which several states gave rise the same number of hypotheses on the final stack, the state on the highest probability path was given precedence. If this failed to break the tie, the state closest to the initial state on the path was selected.

Unfortunately this strategy proved to be less effective than the simpler strategies described previously. We believe the reason may have caused by this strategy selecting states on sets of paths where the best path in the set had too low a rank. We would like to pursue similar ideas in the future, with the overall goal of removing the parameter L_{min} entirely from the decoding process, allowing the decoder more freedom to decode.

5.4. Introducing Segmentation Points into the Stream

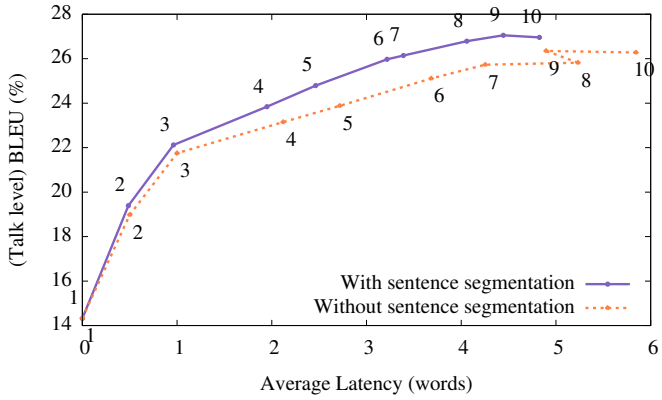
5.4.1. Motivation

As mentioned in Section 2, it has been shown that an input stream can be segmented effectively prior to the decoding process, using information derived from the input word sequence itself (punctuation, part-of-speech tags etc.) and also information from the speech recognition system (for example prosody). In this section we explore the idea of introducing segmentation information into the input stream, to support the segmentation process during stream decoding.

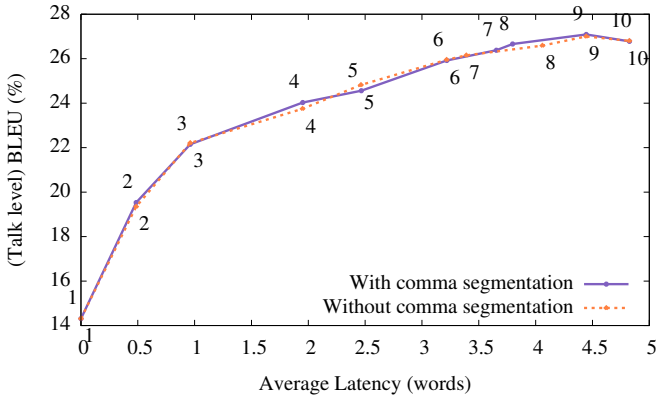
5.4.2. Methodology

In [3] the most effective segmentation strategy was to place segmentation boundaries at commas in the input. In addition segmenting at sentence boundaries also proved to be effective. Using predicted rather than reference commas did not seem to have a negative impact on machine translation performance.

We study the effect of introducing special tokens into the



(a) Using sentence segmentation information.



(b) Using sentence and comma segmentation information.

Figure 7: The effect of introducing segmentation information into the stream for English-to-Spanish.

stream to mark the ends of both sentence internal and sentence final segments. In our experiments we use the positions of commas in the corpus as the position at which to introduce sentence internal segment termination tokens (denoted ⟨p⟩), and the sentence segmentation in the corpus to delimit sentences (using the token ⟨s⟩).

There are a number of plausible strategies for using these tokens during decoding, and we wish to explore more of these in future research. In these experiments we study the case where priority is given to the segmentation indicated by the tokens in the input stream in the following manner: when an ⟨s⟩ or a ⟨p⟩ token arrives on the input stream, the stream decoder translates all untranslated words, and creates an initial search state from which to continue the decoding process. In the case of the ⟨p⟩ token, the language model context is preserved; in the case of ⟨s⟩ it is discarded. In both cases the decoder can violate the L_{min} constraint.

5.4.3. Experiments

The experiments were carried out on data with the punctuation removed from both source and target sides to eliminate the ambiguity of where to place the segmentation tokens in the stream. The punctuation was not used in training the machine translation systems' models, nor was it used in evaluation, but it was used to place the ⟨s⟩ and ⟨p⟩ tokens. The results are shown in Figure 7. It is clear from Figure 7a that sentence boundaries were useful to the stream decoder. The experiment in Figure 7b shows that adding ⟨p⟩ information surprisingly did not give any additional benefit.

6. Conclusions

In this paper we have presented a study of several variations of the stream decoder. The stream decoder is able to decode from a continuous stream of tokens, and is capable of performing segmentation as it decodes. Previous studies have shown this technique can achieve respectable levels of performance whilst maintaining a usefully low level of latency. The experiments in this paper support the original findings and also broaden the study of this decoder by evaluating it on new datasets and new language pairs. Of particular interest were English-Chinese and English-German tasks, which are challenging due to the differences in word order. Our results show that the although BLEU score was impacted at shorter latencies, the behavior of the stream decoder was quite similar in character to that of the language pairs. We believe the original claims that stream decoding can achieve low latency translation with only a small degradation in performance are valid, and can be extended to a broad range of language pairs.

During the course of the research for this paper, we studied a number of alternative strategies for increasing the performance of the decoder. We found a simple but highly effective variant of the stream decoder was one that selected the segmentation point using the n-best list of hypotheses rather than the 1-best. In our experiments this technique substantially improved the performance of the decoder at shorter latencies and also made the decoder less sensitive to the value of the minimum latency constraint.

This paper also proposed a technique for integrating segmentation information from an external source into the stream decoding process. Our experiments show that reliable sentence segmentation information may be used effectively in stream decoding to guide the segmentation process.

In future research we would like to study the behavior of the stream decoder on language pairs with longer distance reordering such as Japanese or Korean to the European languages.

7. References

[1] M. Kolss, S. Vogel, and A. Waibel, "Stream decoding for simultaneous spoken language translation," in *Proceedings of Interspeech*, Brisbane, Australia, 2008.

[2] S. Bangalore, V. K. R. Sridhar, P. Kolan, L. Golipour,

and A. Jimenez, "Real-time incremental speech-to-speech translation of dialogs," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (HLT-NAACL)*, Montreal, Canada, 2012, pp. 437–445.

[3] V. K. R. Sridhar, J. Chen, S. Bangalore, A. Ljolje, and R. Chengalvarayan, "Segmentation strategies for streaming speech translation." in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (HLT-NAACL)*, Atlanta, USA, 2013, pp. 230–238.

[4] Y. Oda, G. Neubig, S. S. T. Toda, and S. Nakamura, "Optimizing segmentation strategies for simultaneous speech translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, USA: The Association for Computer Linguistics, June 2014.

[5] K. Ryu, S. Matsubara, and Y. Inagaki, "Simultaneous English-Japanese spoken language translation based on incremental dependency parsing and transfer," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: The Association for Computer Linguistics, 2006.

[6] P. Padilla, M. T. Bajo, and F. Padilla, "Proposal for a cognitive theory of translation and interpreting," *The Interpreters Newsletter*, 1999.

[7] S. Tirkkonen-Condit, *Tapping and mapping the processes of translation and interpreting: outlooks on empirical research*. John Benjamins Publishing Company, 2000, vol. 37.

[8] E. Matusov, D. Hillard, M. Magimai-Doss, D. Hakkani-Tur, M. Ostendorf, and H. Ney, "Improving speech translation with automatic boundary prediction," in *Proceedings of Interspeech*, Antwerp, 2007, pp. 2449–2452.

[9] M. Kolss, M. Wölfel, F. Kraft, J. Niehues, M. Paulik, and A. Waibel, "Simultaneous German-English lecture translation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Waikiki , Hawai'i, USA, 2008, pp. 174–181.

[10] H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning, "A conditional random field word segmenter for sighan bakeoff 2005," in *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, vol. 171. Jeju Island, Korea, 2005.

[11] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowa, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007): demo and poster sessions*, Prague, Czeck Republic, June 2007, pp. 177–180.

[12] F. J. Och, "Minimum error rate training for statistical machine translation," in *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL 2003)*, Sapporo, Japan, 2003.

[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2001, pp. 311–318.

Incremental Development of Statistical Machine Translation Systems

Li Gong[1,2], Aurélien Max[1,2], François Yvon[1]

LIMSI-CNRS[1] & Univ. Paris-Sud[2]
rue John von Neumann, 91 403 Orsay, France
{firstname.lastname}@limsi.fr

Abstract

Statistical Machine Translation produces results that make it a competitive option in most machine-assisted translation scenarios. However, these good results often come at a very high computational cost and correspond to training regimes which are unfit to many practical contexts, where the ability to adapt to users and domains and to continuously integrate new data (eg. in post-edition contexts) are of primary importance. In this article, we show how these requirements can be met using a strategy for on-demand word alignment and model estimation. Most remarkably, our incremental system development framework is shown to deliver top quality translation performance even in the absence of tuning, and to surpass a strong baseline when performing online tuning. All these results obtained with great computational savings as compared to conventional systems.

1. Introduction

Statistical Machine Translation (SMT) has considerably matured in the past decade and is nowadays a competitive option in most practical machine-assisted translation scenarios. A notable fact about SMT technology is that the construction of high-performance systems is extremely expensive. Even if using appropriate computing resources and parallel programming techniques, building systems for very large data sets requires a significant processing time before any translation can be produced. If individual processing steps may be greatly accelerated, including e.g. word alignment [1] or system tuning [2], the requirement to process the entire parallel data significantly delays the availability of a trained system. And even though a careful pre-selection of bilingual sentences may greatly reduce the size of the training material [3], this selection is itself time-consuming and is not justified when one only needs to translate a handful of documents or documents from multiple domains.

In addition, the trained translation models are *static*. In a state-of-the-art system, all models are extracted from a predefined parallel corpus, and are then used to translate any type of input text. However, new data are constantly made available, and the state-of-the-art SMT approaches cannot seamlessly take advantage of them to improve their performance. Incorporating newly available data can help to increase the n-gram coverage and to improve the parameter estimates of an existing system. These observations provide motivation for incorporating newly available data into existing systems, in particular when the new data is known to be directly relevant to the application documents.

Previous works have empirically shown that not all phrase translation examples are necessary to reach top performance, so that phrase tables can be built on a per-need basis for a given input text using random sampling of translation examples [4, 5]. The main strength of these approaches is that they reduce the computation time of translation models and make it possible to extract translations from very large parallel data, even with arbitrarily long translation units. However, these approaches still require to align all the available parallel data at the word level, a serious bottleneck when working with very large amounts of parallel data.

In this work, we propose to experiment with an architecture where word alignments are only computed on a per-need basis. This proposal naturally enables efficient, *plug-and-play* use of any newly available parallel data, as well as online learning of system parameters. This is similar to the objectives of stream-based SMT [6], but crucially does not require the *actual* alignment of all available data. This means that we are able to develop systems even faster: as our experiments show, immediate integration of newly translated documents, combined with online tuning, make it possible to dispense altogether with the development step. This pragmatic solution offers both the capacity to deliver translations to users much earlier, but also to quickly improve subsequent automatic translations.

The rest of the paper is organized as follows. In Section 2, we describe our framework for efficient on-demand development of SMT systems. We then present in Section 3 experiments designed to demonstrate the capabilities and flexibility of our framework. We finally conclude by reviewing related work in Section 4.

2. On-demand development of SMT systems

2.1. On-the-fly model estimation

A first major difference between our system and a standard SMT pipeline is the ability to compute phrase translation probabilities on a per-need basis, based on small samples of parallel sentences. In our architecture, parallel sentence pairs

are stored in a suffix array [7], enabling fast access to phrase instances.[1] At decoding time, the translation probabilities for all source phrases $\bar{s}$ (up to a given length) are computed based on a subset of occurrences of $\bar{s}$, where the sample size (denoted as M) enables to balance between speed and precision of estimates.

Previous approaches [4, 5] to sampling have resorted to *random deterministic sampling*, which picks a given number of examples by scanning the suffix array index at fixed intervals. The translation probability of a source phrase is then computed as:

$$p(\bar{t}|\bar{s}) = \frac{count(\bar{s}, \bar{t})}{\sum_{\bar{t}'} count(\bar{s}, \bar{t}')} \quad (1)$$

where $count(\cdot)$ is the number of occurrences of the given phrase pair in the sample, which may include occurrences where translation extraction was not possible (what Lopez [5] calls a *coherent* estimation of the translation model, which is found to generally improve performance).

As sampling is performed independently for each source phrase, the computation of the inverse translation probability $p(\bar{s}|\bar{t})$ can no longer be performed exactly. If needed,[2] the following approximation can be used instead:

$$p(\bar{s}|\bar{t}) = min(1.0, \frac{p(\bar{t}|\bar{s}) \times freq(\bar{s})}{freq(\bar{t})}) \quad (2)$$

where $freq(\cdot)$ is the relative frequency of the given phrase in the entire corpus. The numerator ($p(\bar{t}|\bar{s}) \times freq(\bar{s})$) represents the predicted joint probability of $\bar{s}$ and $\bar{t}$.

2.2. On-demand word alignment

The second main peculiarity of our architecture is the ability to perform word alignment on demand for a subset of selected bi-sentences. Word and phrase alignments are required to compute Equation (1), and are obtained using our implementation of the sampling-based alignment method described in [8], which relies on ideas originally introduced in [9]. In this approach, the word alignment between a pair of parallel sentences is generated by a recursive binary segmentation process. Starting with a sentence-level alignment (explicitly available in the parallel corpus), segmentation is performed recursively to match smaller blocks until no block can be further segmented.

This process can be viewed as approximate top-down ITG parsing [10], where matching blocks are determined based on association scores between the words in the source and target sentences. In this study, association scores for the words in the source part of the bi-sentences of interest are generated by a sampling-based transpotting method, which also relies on a sampling strategy and is thus also quite fast. It is however worth noting that any kind of lexical score could be used to measure the strength of word associations.

[1] Querying a suffix array for a phrase of k words can be performed in $(k + \log(|\mathbf{C}|))$ operations, where $|\mathbf{C}|$ is the corpus size. A suffix array can be constructed in $\mathcal{O}(|\mathbf{C}| \log(|\mathbf{C}|))$ time.

[2] Although this model has been shown to be non essential, we use it for the stability of our systems, especially when untuned systems are used.

2.3. System construction

As described before, our framework contains two main parts: on-the-fly model estimation with deterministic random sampling (denoted as `rnd`, henceforth) and on-demand word alignment (denoted as `owa`, henceforth).

The corresponding processing architecture is sketched in Algorithm 1. Given an input document $\mathbf{d}$ to translate, the system first extracts all possible source phrases, $\mathbf{\Sigma}[\mathbf{d}]$. Then, for each source phrase $\bar{s}$ in $\mathbf{\Sigma}[\mathbf{d}]$, we perform deterministic random sampling to select translation examples from the parallel corpus. We then obtain a translation sample of $\bar{s}$, $\mathbf{S}[\bar{s}]$. The sentence pairs in $\mathbf{S}[\bar{s}]$ are then aligned by our on-demand word alignment, where the generated alignments are denoted as $\mathbf{A}_{\mathbf{S}[\bar{s}]}$, and are then used to extract the translations and to compute model parameters $\boldsymbol{\theta}_{\bar{s}}$ for the source phrase $\bar{s}$. This process is repeated for all source phrases in $\mathbf{\Sigma}[\mathbf{d}]$, and the resulting translation table can then be used by a phrase-based decoder to translate the input text into the target language.

Besides the translation models, the other models in our system are the same as in the default configuration of the `moses` system [11], including the lexical weighting and lexicalized reordering models. These models are also computed on-demand based on the computed word alignments.

Algorithm 1 On-demand development procedure

```
Data: training corpus C,
Input: an input document d, sample size M
compute Σ[d]
for all s̄ ∈ Σ[d] do
  S[s̄] = rnd(M, C, s̄)              // Sampling
  A_S[s̄] = owa(S[s̄])               // Alignment
  estimate(θ_s̄, S[s̄], A_S[s̄])     // Estimation
end for
```

3. Experiments

In this section, we have chosen to illustrate two favorable use cases of our framework in order to demonstrate its capabilities and flexibility. The data used in this work is presented in Section 3.1. In Section 3.2, we will use our system in a *translation for communities* task, where documents to be translated are from the same origin, to show its ability to quickly adapt to a specific domain and take advantage of similarities between documents to outperform a strong baseline. In Section 3.3, another even more difficult use case, which we called *any-text translation*, will be studied.

3.1. Data

We selected English-French as our main language pair for this study, mostly because large quantities of parallel data

Documents	# lines	#token$_{en}$	#token$_{fr}$	Domains
WMT	16.6M	396.9M	475.1M	Mixture
Cochrane(dev)	743	16.5K	21.4K	Medical
Cochrane(100 docs)	1.8K	38.6K	49.3K	Medical
talk1	232	4.2 K	4.3 K	TedTalk
talk2	249	5.2 K	5.9 K	TedTalk
book1	1093	22.5 K	23.8 K	Literature
book2	1604	35.1 K	37.8 K	Literature
subtitle1	495	5.0 K	5.6 K	Open subtitle
subtitle2	528	4.8 K	5.2 K	Open subtitle
php	1000	11.6 K	12.5 K	Technical manual
kdedoc	995	11.8K	12.5 K	Technical manual

Table 1: Description of corpora used in our experiments.

Configs	Translation quality		PT construction time	
	BLEU	TER	user CPU	wall clock
moses	34.12	48.59	1,212h	252h
on-demand	28.58	49.54	76h	7h
+spec	32.33	46.42	76.5h	7h
+online	36.41	46.44	76.5h	7h
+dev	36.20	46.10	148.5h	14h

Table 2: Results for the owa system on a large-scale English-to-French translation task.

are readily available for this language pair. Data from the Workshop on Statistical Machine Translation (WMT)[3] from a variety of domains were used, as well as additional data from various origins from the medical domain and used in the WMT'14 medical task.[4] This dataset, denoted as WMT, contains data from different domains, including News commentaries, parliamentary debates and medical texts.

In the "translation for communities" scenario, we used data of systematic summaries for specialists from the Cochrane collaboration.[5] The Cochrane dataset is made up of short documents typically containing one or two dozen of sentences. In the "any-text translation" scenario, we chose 8 documents from various domains: two entire transcriptions of TED Talks, two translated books, two movie subtitles and two technical manuals. Table 1 provides basic statistics regarding these corpora. Tokenization was performed using in-house tools.

3.2. Translation for communities

In the translation for communities task, we make two important assumptions: the first one is that it can be desirable to provide automatic translations early, even before any human translation has been performed, to handle *documents of unknown origin so far* (as is the case when a new application domain is considered); the second one is that there exists some clear relation between consecutive application documents, so that their set of optimal parameters are close to one another. A consequence of these assumptions is that a classical development set will not be needed anymore, a significant economy in practice. Nonetheless, our proposal only makes sense if it also compares favorably in terms of translation evaluation to a standard system making use of a development set.

We thus constructed a vanilla moses system. We used mgiza++[6] to align the full bi-corpus and the moses scripts to extract a huge phrase table and a reordering table for the entire parallel corpus (respectively 20Gb and 7.5Gb compressed on disk), which have to be filtered for each input text. The medical-domain LM was trained on the French side of WMT'14 medical data (containing 4.8M sentences and 78M tokens). The system was optimized with KBMIRA, a variant of the Margin Infused Relaxation Algorithm described in [12], on the Cochrane development set. Translations are computed with the moses phrase-based decoder. Results are reported using the BLEU [13] and TER [14] metrics.

In this first scenario, we consider a situation where a stream of documents needs to be translated. After each document has been automatically processed, we also make the plausible assumption that it is post-edited by a human translator, thus providing new data that can be used to update both the models and parameters of the systems before translating the next document.

This situation is illustrated using the Cochrane dataset, where we take the 100 documents constituting the test set (see Table 1) to simulate the document stream. In the following, we describe a series of increasingly rich configurations and show that our framework can deliver fast, yet competitive translations for these documents.

3.2.1. *On-demand development of systems (on-demand)*

In the first configuration, our system processes each input document separately in sequence, as described in Algorithm 1. Word alignments of previously aligned sentences will be cached and readily be available for subsequent documents. Each document-specific translation table is fed to the decoder[7], which uses the default values for all model parameters. In this configuration, no tuning is actually performed, which eliminates completely the need for a development corpus and allows us to obtain translation of documents almost instantly.[8]

Results for this untuned configuration (see on-demand in Table 2) are lower by 5.5 BLEU point (BP) than those of the conventionally tuned moses system, which can be mostly attributed to the absence of tuning. However, translations for the test set are delivered much faster, where our system is x36 times wall clock faster than moses.

[3] http://www.statmt.org/wmt13

[4] http://www.statmt.org/wmt14

[5] http://summaries.cochrane.org

[6] http://www.kyloo.net/software/doku.php/mgiza:overview

[7] We used the moses decoder in our experiments, whose default parameters are: 0.3 for all 7 reordering features, including 6 lexical reordering features and 1 distance-based reordering feature; 0.2 for all 5 translation features; 0.5 for the language model and -1 for the word penalty.

[8] In this work, the language model is still pre-trained. Future work will include the incremental / on-demand estimation of language models [15].

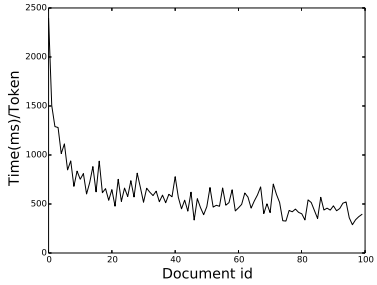


Figure 1: Evolution of the average per token processing time for a sequence of documents.

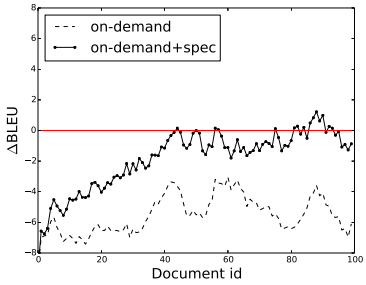


Figure 2: Document-level comparison with `moses` system in English-to-French translation direction. The y-axis represents the difference in BLEU score (ΔBLEU) between our systems and the vanilla `moses` system for each document in the sequence.

As mentioned before, the computed word alignments are cached and are available for translating subsequent documents. To further analyze the effect of the cache, Figure 1 shows how the average per token processing time decreases as more and more documents from the same flow are translated. At the outset, estimation time per token decreases quickly as a result of the use of the cache; as more and more documents are translated, the average estimation time continues to decrease, albeit at a slower pace.

3.2.2. Plug-and-play data integration (`+spec`)

We now consider the following *incremental training* regime: after each individual document is translated, the post-edited version of the document becomes available.[9] Our on-demand framework makes it natural and straightforward to integrate any newly available parallel data without any full retraining.

In the following experiment, each newly available `Cochrane` document is added to a "specialized" corpus, denoted by `spec`. A separate phrase table for each document is estimated from `spec` using Algorithm 1; considering the very small size of our specialized source, the corresponding phrase table, built from previous documents in the sequence $\{\mathbf{d}_i, i = 1 \dots t-1\}$, contains only two scores per phrase pair: the direct translation model score and the phrase penalty. As we still assume that no development set is available, the parameters for the new models are being thus simply copied from the main table. Note that in this setting, the `spec` phrase table is used as a *back-off* table to the phrase table estimated from the main, static corpus. While this may seem counter-intuitive, we did this primarily because the `spec` translation model is comparatively poorly estimated, because of the small quantity of data used. However, for those domain-specific terms, phraseology or long phrases which usually only exist in the in-domain data, we could use the `spec` phrase table to translate them.

[9] In fact, the `Cochrane` dataset used in this study is made of two parts: a large portion of the data was translated by human translator from scratch, while a smaller amount a document where actually produced through post-edition. We still use this data as a post-edited corpus in our experiments, although these two kinds of data are slightly different. We believe this does not affect our experimental conclusions [16].

Results in Table 2 show that the additional table (`+spec`) helps to significantly improve translation quality over the raw `on-demand` configuration (+3.7 BP), for a modest additional processing time of half an hour for aligning the content of the first 99 documents. Since the `spec` table for document $\mathbf{d}_t$ is estimated based on the previous $t-1$ documents, the quality of the phrase table improves over time.

Figure 2 shows the document-level comparison between our systems (`on-demand` and `on-demand+spec`) and the vanilla `moses` system, where the curves represents the difference of performance (evaluated by BLEU) between `moses` and the corresponding system on each document in the stream. The parts above the horizontal line means the corresponding system is better than `moses`; otherwise, the corresponding system is worse. We first observe that the document-level gap between `on-demand` and `on-demand+spec` is much larger (around 5 BP) at the end of the document sequence than at the start, confirming that the quality of the `spec` phrase table improves over time. We also see that `on-demand` systematically underperforms `moses` on all documents, which was expected given the gap in corpus-level performance. Interestingly, the use of the specialized phrase table, `on-demand+spec`, yields fast improvements and matches the performance of `moses` after about 40 documents have been translated. We can conclude that the integration of such a specialized corpus allows our system to achieve nearly the same performance as the vanilla `moses` system but delivering translations much faster. Furthermore, these results are obtained without using a development set, a significant economy both in human translation time and in system development time. Although the obtained results strongly depend on the nature of the data used, the *plug-and-play* data integration feature of our framework is very useful to improve the translation performance when translating streams of related documents.

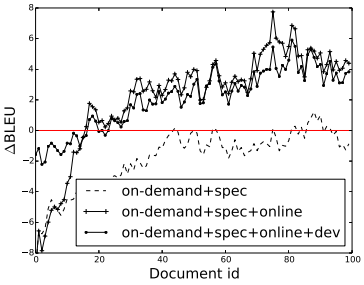


Figure 3: Document-level comparison with `moses` system in English-to-French translation direction. Initialization either uses `moses` default values (`+online`), or parameters tuned on a development set (`+dev`).

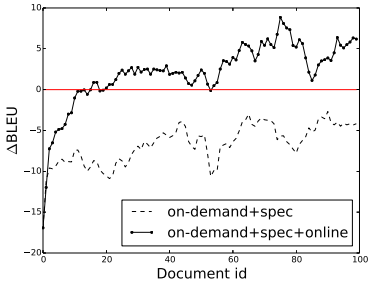


Figure 4: Document-level comparison with `moses` system in French-to-English translation direction.

3.2.3. Simple online tuning (`+online`)

We have previously shown that our on-demand framework allows us to seamlessly integrate newly available data, yielding systems that match a `moses` system trained in a conventional way after just 40 documents of our specific data source. Remarkably, these results were obtained *without any parameters tuning*. We now consider a simple online tuning strategy to further explore the potential of on-demand system development. In practice, the system's weights are retuned after each document has been translated (and post-edited) as follows: Taking the previous weights as the initial point, we run the parameter tuning process (here `KBMIRA`) on the just translated and post-edited document; the resulting parameter values are then averaged with the parameter values of the 10 previous documents[10], and then used for translating the next document. Additionally, in order to leverage the `spec` table, we also allow here the `spec` phrase table to compete with the phrase table estimated from the static corpus [17] instead of having the latter take precedence.

Results for this last configuration are given in Table 2 (`+online`). Our simple online tuning yields a significant improvement (+4.1 BP) over the untuned `on-demand+spec` configuration. Even though the two configurations cannot be directly compared at the corpus-level, since our system integrates a growing set of in-domain data, while `moses` on its part greatly benefits from the in-domain development data, we still note that our framework now outperforms the `moses` baseline (+2.3 BP). More interestingly, comparison at the document-level (see Figure 3) demonstrates the strong potential of our framework: `moses` is systematically outperformed after fewer than 20 documents are translated. As for processing time, documents being very small, online tuning only takes 3mn (wall clock time) on average for each document in this experiment.

Our final experiment in the translation for communities scenario is designed to analyze the performance of our last configuration if it starts with conventionally tuned initial parameters. We thus first tuned the system on the development set, and then used the tuned parameters to initialize the starting parameters of this new configuration. The result is reported in Table 2 (`+dev`): using tuned parameters to initialize the system yields no significant change on translation quality. Comparing to the `on-demand+spec+online` system, BLEU by 0.2 points but TER is better by 0.3 points. The document-level comparison in Figure 3 shows, as expected, that initializing with parameters tuned on the development set yields better performance than `on-demand+spec+online` at the start of the document sequence. However, after fewer than 20 documents have been processed, there is no visible difference between the two systems. We can thus conclude that the online tuning strategy implemented in our framework allows us to effectively dispense with the use of a development set.

Finally, we also performed these experiments on the French-to-English translation direction, and the corresponding document-level results are shown in Figure 4. First, for the `on-demand+spec` system, we observe that the performance of the system improves with the number of translated documents, although it is not as significant as the improvement observed in the English-to-French translation direction (as shown in Figure 3). This is probably related to the diversity of the language: indeed, the 100 `Cochrane` bilingual documents contain 3 854 unique English words and 4 398 unique French words. A larger vocabulary implies a lower repetition rate, which makes `+spec` less beneficial. When applying online tuning, our best system (`on-demand+spec+online`) again improves quickly and outperforms the `moses` baseline after less than 20 documents have been translated.

3.3. Any-text translation

In this section, we consider a comparatively less studied, albeit somewhat more realistic, scenario, where the characteristics of the input text are completely unknown before translation. We thus make the following assumptions:

- Training data was collected opportunistically and no specific document metadata (e.g. genre, document

[10] We restrict to the more recent documents to make tuning more reactive to changes in the quality of the `spec` table.

boundaries) are available for the full data set.

- The input text corresponds to a coherent discourse (i.e. is not made by concatenating unrelated documents).
- The text can be from any arbitrary domain, which precludes any off-line adaptation using a predefined specific bilingual corpora.
- No adapted development set is available, which precludes the use of tuning techniques relying on a development corpus from the same data source or domain.

Since the input text is completely unknown and could be from any domain, we dub this scenario *any-text translation*.

As presented above, experiments are performed on 8 documents from various domains (see Table 1). Each document is translated independently, sentence by sentence. Translation rules are extracted from the training corpus for each sentence using an adapted version of Algorithm 1, where each sentence is treated as a single document.

We also make the same assumption as in Section 3.2 that after each sentence has been automatically translated, a reference translation is made available by a human translator (simulating a post-edition scenario, even though the documents used in this section have not been post-edited). These translated and reference data are used to update both the models and parameters for the next sentences. In this study, each sentence is translated with two phrase tables: one is estimated based on the training data of the system, the other is estimated based on the previously translated sentences in the same document (denoted by `indoc`[11]).

Again, we chose the large-scale corpus `WMT` (see Table 1) as the training data and the vanilla `moses` system as our baseline. Since no development set is available, we chose to use the decoder's default parameters as initial parameters for decoding. As for the target language model, a general-domain LM was used which was trained on the `WMT` corpus. Since the `WMT` corpus contains very large quantities of data from different domains, this LM could be considered as a reasonable general-domain LM.

Experimental results are presented in Table 3, where `moses` is the baseline system, `on-demand` represents our on-demand SMT system, and `+indoc` represents our on-demand SMT system but also using the `indoc` phrase table in decoding. First, by comparing the results of `moses` and `on-demand` systems, we find `moses` is better than `on-demand` on all documents on BLEU. On TER, `moses` is also better than `on-demand` on most documents (5 out of 8 documents). We attribute this result to the effect of sampling and the differences in word alignments: our models are estimated based on a subset of translation examples while the models in `moses` system are estimated based on all examples in the corpus and our on-demand word alignments are probably a little worse than the `mgiza++` word

[11] Actually, `indoc` is similar to previous spec phrase table, but `indoc` is estimated based on translated data in the same document.

Documents	Baseline		Systems			
	moses		on-demand		+indoc	
	BLEU	TER	BLEU	TER	BLEU	TER
talk1	27.84	56.99	27.27	57.34	**28.30**	56.53
talk2	**30.96**	50.20	29.13	50.88	29.08	50.94
book1	15.29	68.64	14.87	67.93	**17.12**	65.56
book2	14.71	69.21	13.84	69.39	**14.75**	68.23
subtitle1	**25.10**	56.44	24.25	55.69	24.41	55.30
subtitle2	**29.79**	49.85	29.05	49.96	29.72	49.60
php	17.42	66.24	16.43	67.38	**25.17**	60.96
kdedoc	11.02	82.09	10.08	80.16	**13.43**	77.47

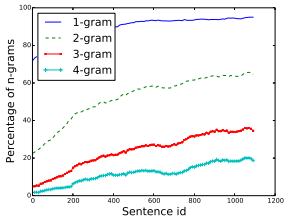
Table 3: Any-text machine translation results for English-to-French translation.

alignments on large-scale corpora. Second, by adding the `indoc` phrase table, our `on-demand` systems (`+indoc`) are generally improved, except on `talk2`, and they are better than `moses` for BLEU on most documents (6 out of 8 documents). Apparently, such improvements depend on the repetitiveness and the length of documents.

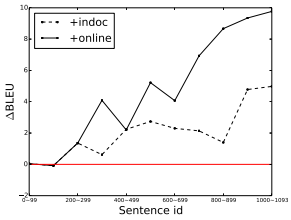
In this use case, it is also possible to perform parameter tuning during the translation of individual documents. Unlike the situation in Section 3.2, where the translation unit was the document, here one sentence contains too little information to perform parameter tuning. Hence, instead, we chose to perform parameter tuning after small batches (of size 100 in our experiments) have been translated. In this experiment, the first sentences of a document are always translated using the decoder's default parameters. After each has batch been translated, the corresponding references are made available and used as a development set to tune the parameters, again with `KBMIRA`. The updated parameters are then used to translate subsequent sentences. In order to assess the effect of parameter tuning on translation results, we only apply the tuning process to a few long documents (> 1000 sentences): `book1`, `book2` and `php`.

For `book1`, applying parameter tuning after each group of 100 sentences for the `+indoc` system yields a further improvement of $+2.4$ BP and -0.1 TP. On `book2`, the result is less clear: the BLEU score is improved by $+0.3$ BP comparing to the `+indoc` system, but the TER score becomes worse by $+1.8$ TP. On the `php` document, a significant improvement from the `+indoc` is observed ($+9.2$ BP, -4.6 TP).

To better understand the behavior of our system, we also performed document-level analyses on these results. Figure 5a shows the percentage of n-grams occurring in sentence $\mathbf{s}_t$ that were also seen in the previous $t-1$ sentences $\{\mathbf{s}_i, i = 1 \dots t-1\}$. For instance, for the sentences at the end of `book1`, about 20% of 4-grams (and nearly 40% of 3-grams) were found in the previous sentences of the document. Figure 5b shows the BLEU scores estimated on each group of 100 sentences. In the `+indoc` system, all sentences are decoded with the default parameters of `moses`, while in the `+online` system, the decoder parameters for each group

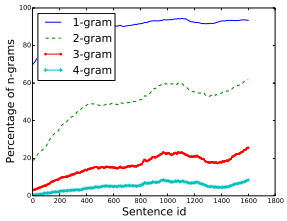


(a) n-gram repetition

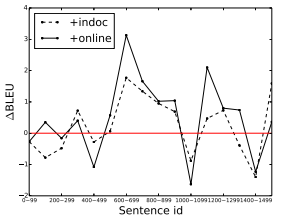


(b) comparison with `moses`

Figure 5: Experimental results on `book1`.



(a) n-gram repetition



(b) comparison with `moses`

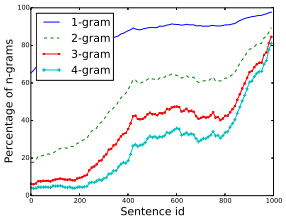
Figure 6: Experimental results on `book2`.

of 100 sentences are tuned on the previous 100 sentences.[12] As shown in Figure 5b, the `+indoc` system takes advantage of the repetitiveness of the document and its performance is systematically better than `moses` after translating 200 sentences. By applying parameter tuning on each group of 100 sentences, results are further improved, and to a larger extent (about 5 BP) at the end of the document.

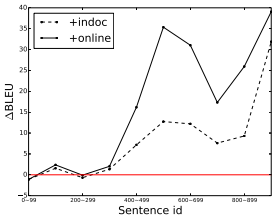
Now turning to `book2`, we find that the results are very different than for `book1`. First, as shown in Figure 6a, the n-gram repetition rate is lower than that of `book1`, especially for 3-grams and 4-grams. For instance, less than 10% of the 4-grams occurring in sentences at the end of the document, were seen in previous passages. The effect of the low repetitiveness of the document is also reflected on the corpus-level evaluation (see Table 3), where adding the `indoc` phrase table only improves performance by +0.9 BP, which compares poorly with the (+2.2 BP) improvement observed for `book1`. In this situation, parameter tuning does not always improve translation performance (only in 11 out of 15 sentence groups), and sometimes even proves detrimental to translation quality (see Figure 6b). This result may be related to overfitting issues and suggests to use more sophisticated online adaptation strategies.

Finally, on `php`, the results are much clearer. As shown in Figure 7a, the `php` document has a very high repetition rate. The effect of such a high repetition rate is directly reflected on the translation results shown in Figure 7b, where the `+indoc` system improves very quickly along with the

[12] For example, the sentences 201 to 300 are decoded with the parameters which are tuned on the sentences 101 to 200.



(a) n-gram repetition



(b) comparison with `moses`

Figure 7: Experimental results on the `php` document.

number of translated sentences, and the improvement is very large. With tuned parameters, the system could better take advantage of the `indoc` phrase table, and the results are further improved.

In this series of experiments, we have demonstrated that our framework can quickly construct SMT systems and incrementally adapt them to the target domain, even though the input texts are completely unknown. Its on-demand training character makes it possible to immediately produce translation output, even though the translation quality at the beginning is not very competitive. Also, its incremental adaptation scheme quickly improves its performance, especially on long and repetitive documents.

4. Related Work

Our framework provides an innovative methodology that is also suitable for interactive MT: we measured wall clock times of less than 1 minute (*before any cache is available*) to build translation tables for individual sentences, making it practical to integrate system development within interactive human post-editing.

Interactive Machine Translation (IMT) was pioneered by projects such as TransType [18], where an SMT system assists the human translator by proposing translation completions that the translator can accept, modify or ignore. IMT was later further developed to enable more types of interaction [19, 20] and to integrate the result of the interaction to influence future choices of the system. More recently, online learning was introduced in the IMT framework [21] to improve the exploitation of the translator's feedback.

A similar idea was also presented in [22]. In this work, the input document is processed sentence by sentence. After the translation of each sentence, the MT output and the post-edited translation are analyzed and used to extract post-editing rules. These rules are then used to automatically process the MT output so as to improve the quality of output translations.

5. Conclusion

This work has addressed the issue of how the computationally expensive cost of the development of high-performance

SMT systems, which typically exploit very large quantities of data, can be significantly reduced. By using our incremental strategies, reductions of computation time up to 36 times were obtained relative to a state-of-the-art system trained in a conventional fashion. Fast integration of newly available data in conjunction with online tuning allowed us to quickly reach the same performance as a strong baseline.

We lastly want to underline that scenarios based on the `+spec` characteristic make simpler assumptions than traditional interactive MT (e.g. [18, 19, 21]), as parameter updates are synced to the stream of incoming documents. In addition, as illustrated in Section 3.3, the on-demand strategy is also capable to perform the more fine-grained scenario of interactive MT, with the distinguishing characteristics that the MT system does *not even need to exist* before its actual use.

6. References

[1] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *Proceedings of NAACL-HLT*, Atlanta, USA, 2013, pp. 644–648.

[2] S. Green, S. Wang, D. Cer, and C. D. Manning, "Fast and adaptive online training of feature-rich translation models," in *Proceedings of ACL*, Sofia, Bulgaria, 2013, pp. 311–321.

[3] G. Gascó, M.-A. Rocha, G. Sanchis-Trilles, J. Andrés-Ferrer, and F. Casacuberta, "Does more data always yield better translations?" in *Proceedings of EACL*, Avignon, France, 2012, pp. 152–161.

[4] C. Callison-Burch, C. Bannard, and J. Schroeder, "Scaling phrase-based statistical machine translation to larger corpora and longer phrases," in *Proceedings of ACL*, Ann Arbor, USA, 2005, pp. 255–262.

[5] A. Lopez, "Tera-scale translation models via pattern matching," in *Proceedings of COLING*, Manchester, UK, 2008, pp. 505–512.

[6] A. Levenberg, C. Callison-Burch, and M. Osborne, "Stream-based translation models for statistical machine translation," in *Proceedings of HLT-NAACL*, Los Angeles, USA, 2010, pp. 394–402.

[7] U. Manber and G. Myers, "Suffix arrays: A new method for on-line string searches," in *Proceedings of SODA*, Philadelphia, USA, 1990, pp. 319–327.

[8] L. Gong, A. Max, and F. Yvon, "Improving bilingual sub-sentential alignment by sampling-based transpotting," in *Proceedings of IWSLT*, Heidelberg, Germany, 2013.

[9] A. Lardilleux, F. Yvon, and Y. Lepage, "Hierarchical Sub-sentential Alignment with Anymalign," in *Proceedings of EAMT*, Trento, Italy, 2012, pp. 280–286.

[10] D. Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," *Computational Linguistics*, vol. 23, no. 3, pp. 377–403, 1997.

[11] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of ACL*, Prague, Czech Republic, 2007, pp. 177–180.

[12] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of NAACL-HLT*, Montréal, Canada, 2012, pp. 427–436.

[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of ACL*, Philadelphia, USA, 2002, pp. 311–318.

[14] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of AMTA*, Boston, USA, 2006, pp. 223–231.

[15] A. Levenberg and M. Osborne, "Stream-based randomised language models for SMT," in *Proceedings of EMNLP*, Singapore, 2009, pp. 756–764.

[16] M. Denkowski, C. Dyer, and A. Lavie, "Learning from post-editing: Online model adaptation for statistical machine translation," in *Proceedings of EACL*, Gothenburg, Sweden, 2014, pp. 395–404.

[17] P. Koehn and J. Schroeder, "Experiments in domain adaptation for statistical machine translation," in *Proceedings of WMT*, Prague, Czech Republic, 2007, pp. 224–227.

[18] P. Langlais, G. Foster, and G. Lapalme, "TransType: a computer-aided translation typing system," in *Proceedings of NAACL-ANLP*, Seattle, USA, 2000, pp. 46–51.

[19] S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, and J.-M. Vilar, "Statistical approaches to computer-assisted translation," *Computational Linguistics*, vol. 35, no. 1, pp. 3–28, Mar. 2009.

[20] P. Koehn, "A web-based interactive computer aided translation tool," in *Proceedings of the ACL-IJCNLP Software Demonstrations*, Singapore, 2009, pp. 17–20.

[21] D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta, "Online learning for interactive statistical machine translation," in *Proceedings of HLT-NAACL*, Los Angeles, USA, 2010, pp. 546–554.

[22] M. Simard and G. Foster, "Pepr: Postedit propagation using phrase-based statistical machine translation," in *Proceedings of MT Summit*, Nice, France, 2013, pp. 191–198.

Lexical Translation Model Using A Deep Neural Network Architecture

Thanh-Le Ha, Jan Niehues, Alex Waibel

International Center for Advanced Communication Technologies - InterACT
Institute of Anthropomatics and Robotics
Karlsruhe Institute of Technology, Germany
{thanh-le.ha|jan.niehues|alex.waibel}@kit.edu

Abstract

In this paper we combine the advantages of a model using global source sentence contexts, the Discriminative Word Lexicon, and neural networks. By using deep neural networks instead of the linear maximum entropy model in the Discriminative Word Lexicon models, we are able to leverage dependencies between different source words due to the non-linearity. Furthermore, the models for different target words can share parameters and therefore data sparsity problems are effectively reduced.

By using this approach in a state-of-the-art translation system, we can improve the performance by up to 0.5 BLEU points for three different language pairs on the TED translation task.

1. Introduction

Since the first attempt to statisical machine translation (SMT) [1], the approach has drawn much interest in the research community and huge improvements in translation quality have been achieved. Still, there are plenty of problems in SMT which should be addressed. One is that the translation decision depends on a quite small context.

In standard phrase-based statistical machine translation (PBMT) [2], the two main components are the translation and language models. The translation model is modeled by counting phrase pairs, which are sequences of words extracted from bilingual corpora. By using phrase segments instead of words, PBMT can exploit some local source and target contexts within those segments. But no context information outside the phrase pairs is used. In an n-gram language model, only a context of up to n target words is considered.

Several directions have been proposed to leverage information from wider contexts in the phrase-based SMT framework. For example, the Discriminative Word Lexicon (DWL) [3][4] exploits the occurence of all the words in the whole source sentence to predict the presence of words in the target sentence. This wider context information is encoded as features and employed in a discriminative framework. Hence, they train a maximum entropy (MaxEnt) model for each target word.

While this model can improve the translation quality in different conditions, MaxEnt models are linear classifiers. On the other hand, hierarchical non-linear classifiers can model dependencies between different source words better since they perform some abstraction over the input. Hence, introducing non-linearity into the modeling of the lexical translation could improve the quality. Moreover, since many pairs of source and target words co-occur only rarely, a way of sharing information between the different classifiers could improve the modeling as well.

In order to address these issues, we developed a discriminative lexical model based on deep neural networks. Since we train one neural network for all target words as a multivariate binary classifier, the model can share information between different target words. Furthermore, the probability is no longer a linear combination of weights depending on the surface source words. Thanks to the non-linearity, we are now able to exploit semantic dependencies among source words.

This paper is organized as follows. In Section 2, we review the previous works related to lexical translation methods as well as the translation modeling using neural networks. Then we describe our approach including the network architecture and its training procedures in Section 3. Section 4 provides experimental results of our translation systems for different language pairs using the proposed lexical translation model. Finally, the conclusions are drawn in Section 5.

2. Related work

Since the beginnings of SMT, several approaches to increase the context used for lexical decisions have been presented. When moving from word-based to phrase-based SMT [2][5], a big step in employing wider contexts into translation systems has been made. In PBMT, the lexical joint models allow us to use local source and target contexts in the form of phrases. Lately, advanced joint models have been proposed to either enhance the joint probability model between source and target sides or engage more suitable contexts.

The n-gram based approach [6] directly models the joint probability of source and target sentences from the conditional probability of a current n-gram pair givens sequences

of previous bilingual n-grams. In [7], this idea is introduced into the phrase-based MT approach. Thereby, parallel context over phrase boundaries can be used during the translation.

Standard phrase-based or n-gram translation models are basically built upon statistical principles such as Maximum Entropy and smoothing techniques. Recently, joint models are learned using neural networks where non-linear translation relationships and semantic generalization of words can be performed [8]. Le et. al. [9] follow the n-gram translation direction but model the conditional probability of a target word given the history of bilingual phrase pairs using a neural network architecture. They then use their model in a k-best rescorer instead of in their n-gram decoder. Devlin et. al. [10] add longer source contexts and renew the joint formula so that it can be included in a decoder rather than a k-best rescoring module. Schwenk et. al. [11] calculate the conditional probability of a target phrase instead of a target word given a source phrase.

Although the aforementioned works essentially augment the joint translation model, they have an inherent limitation: only exploit local contexts. They estimate the joint model using sequences of words as the basic unit. On the other hand, there are several approaches utilizing global contexts. Motivated by Bangalore et. al [12], Hasan et. al. [13] calculate the probability of a target word given two source words which do not necessarily belong to a phrase. Mauser et. al. [3] suggest another lexical translation approach, named Discriminative Word Lexicon (DWL), concentrating on predicting the presence of target words given the source words. Niehues et. al. [4] extend the model to employ the source and target contexts, but they used the same MaxEnt classifier for the task. Carpuat et. al. [14] is the most similar work to the DWL direction in terms of using the whole source sentence to perform the lexical choices of target words. They treat the selection process as a Word Sense Disambiguation (WSD) task, where target words or phrases are WSD senses. They extract a rich feature set from the source sentences, including source words, and input them into a WSD classifier. Still, the problem persists since they use the shallow classifiers for that task.

Considering the advantages of non-linear models mentioned before, we opt for using deep neural network architectures to learn the DWL. We take the advantages of the two directions. On one side, our model uses a non-linear classification method to leverage dependencies between different source sentences as well as its semantic generalization ability. On the other side, by employing the global contexts, our model can complement joint translation models which use the local contexts.

3. Discriminative lexical translation using deep neural networks

We will first review the original DWL approach described in [3] and [4]. Afterwards, we will describe the neural network architecture and training procedures proposed in this work. We will finish this section by describing the integration into the decoding process.

3.1. Original Discriminative Word Lexicon

In this approach, the DWL are modeled using a maximum entropy model to determine the probability of using a target word in the translation. Therefore, individual models for every target word are trained. Each model is trained to return the probability of this word given the input sentence.

The input of the model is the source sentence, thus, they need a way to represent the input sentence. This is done by representing the sentence as a bag of words and thereby ignoring the order of the words. In the MaxEnt model, they use an indicator feature for every input word. More formally, a given source sentence $s = s_1 \dots s_I$ is represented by the features $F(s) = \{f_w(s) : \forall w \in V_s\}$, with V_s is the source vocabulary:

$$f_w(s) = \begin{cases} 1 & \text{if} \quad w \in s \\ 0 & \text{if} \quad w \notin s \end{cases} \tag{1}$$

The models are trained on examples generated by the parallel training data. The labels for training the classifier of target word t_j are defined as follows:

$$label_{t_j}(s,t) = \begin{cases} 1 & \text{if} \quad t_j \in t \\ 0 & \text{if} \quad t_j \notin t \end{cases} \tag{2}$$

This model approximates the probability $p(t_j|s)$ of a target word t_j given the source sentence s. We will discuss our alternative method using neural network to estimate those probabilities in the next section.

In [4], the source context is considered in a way that the sentence is no longer represented by a bag of words, but by a bag of ngrams. Using this representation, they could integrate the order information of the words, but the dimension of the input space is increased. We also adapt this extension to our model by encoding the bigrams and trigrams as ordinary words in the source vocabulary.

After inducing the probability for every word t_j given the source sentence s, these probabilities were combined into the probability of the whole target sentence $t = t_1 \dots t_J$ given s as described in Section 3.4.

3.2. General network architecture

After we reviewed the original DWL in the last section, we will now describe the neural network that replaces the MaxEnt model for calculating the probabilities $p(t_j|s)$.

The input and output of our neural network-based DWL are the source and target sentences from which we would like to learn the lexical translation relationship. As in the original DWL approach, we represent each source sentence s as a binary column vector $\hat{\mathbf{s}} \in \{0|1\}^{|V_s|}$ with V_s being the considered vocabulary of the source corpus. If a source word s_i

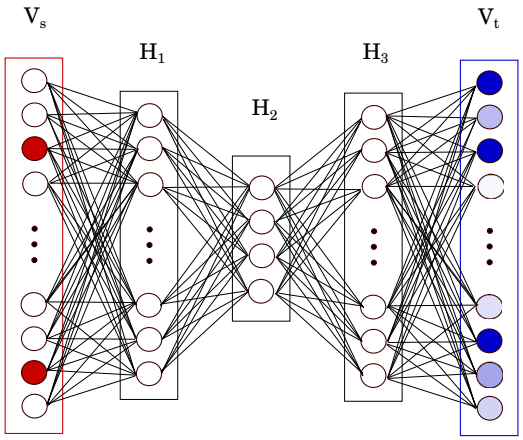


Figure 1: *FFNN architecture for learning lexical translation.*

appears in that sentence s, the value of the corresponding index i in $\hat{\mathbf{s}}$ is 1, and 0 otherwise. Hence, the source sentence representation should be a sparse vector, depending on the considered vocabulary V_s. The same representation scheme is applied to the target sentence t to get a sparse binary column vector $\hat{\mathbf{t}}$ with the considered target vocabulary V_t.

As the Figure 1 depicts, our main neural network-based DWL architecture for learning lexical translation is a feed-forward neural network (FFNN) with three hidden layers. The matrix $\mathbf{W}^{(1)} \in \mathbb{R}^{V_s \times |H_1|}$ connects the input layer to the first hidden layer. Two matrices $\mathbf{W}^{(2)} \in \mathbb{R}^{|H_1| \times |H_2|}$ and $\mathbf{W}^{(3)} \in \mathbb{R}^{|H_2| \times |H_3|}$ encodes the learned translation mapping between two compact global feature spaces of the source and target contexts. And the matrix $\mathbf{W}^{(4)} \in \mathbb{R}^{|H_3| \times |V_t|}$ computes the lexical translation output. $|H_1|$, $|H_2|$, and $|H_3|$ are the number of units in the first, second and third hidden layers, respectively. The lexical translation distribution of the words in the target sentence $p(t_i|s)$ for a given source sentence s is computed by a forward pass:

$$p(t_i|s) = \sigma_i({\mathbf{W}^{(4)}}^T \mathbf{O}^{(3)})$$

where:

$$\mathbf{O}^{(k)} = \left[\sigma_j({\mathbf{W}^{(k)}}^T \mathbf{O}^{(k-1)})\right] \quad k \in \{1, 2, 3\}$$
$$\mathbf{O}^{(0)} = \hat{\mathbf{s}} \quad \text{and} \quad \mathbf{O}^{(4)} = \mathbf{p}(t, s)$$

and σ_j is the *sigmoid* function $\sigma(x)$ applied to the j^{th} value in a column vector:

$$\sigma(x) = \frac{1}{1 + \mathbf{e}^{-x}}$$

So the parameters of the network are:

$$\theta = (\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}, \mathbf{W}^{(4)})$$

To investigate the impact of the network configuration, we built a simpler architecture with only one hidden layer featuring the translation relationship between source and target sentences. We will refer this as the *SimNNDWL* in the comparison section later.

3.3. Network training

In neural network training, for each instance, which is comprised of a sentence pair (s, t), we maximize the similarity between the conditional probability $p_i = p_\theta(t_i|s)$ to either 1 or 0 depending on the appearance of the corresponding word t_i in the target sentence t. The neural network operates as a multivariate classifier which gives the probabilistic score for a binary decision of independent variables, i.e the appearances of target words. Here we minimize the cross entropy error function between the binary target sentence vector $\hat{\mathbf{t}}$ and the output of the network $\mathbf{p} = [p_i]$:

$$E = -\frac{1}{V_t} \sum_{i=1}^{V_t} (\hat{\mathbf{t}}_i \ln p_i + (1 - \hat{\mathbf{t}}_i) \ln(1 - p_i))$$

We train the network by back-propagating the error based on the gradient descent principle. The error gradient for the weights between the last layer and the output is calculated as:

$$\frac{\partial E}{\partial w_{ij}^{(4)}} = (\mathbf{O}_j^{(4)} - \hat{\mathbf{t}}_j)\mathbf{O}_i^{(3)}$$

The error gradient for the weights between the other layers is calculated based on the error gradients for activation values from the previous layers:

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = \frac{\partial E}{\partial \mathbf{O}_j^{(k)}} \mathbf{O}_i^{(k-1)}$$

Then the weight matrices are batch-updated after each epoch:

$$\mathbf{W}^{(k)}[T+1] = \mathbf{W}^{(k)}[T] - \eta \sum_{i=1}^{N} \frac{\partial E}{\partial \mathbf{W}^{(k)}}$$

where:

- N is the number of training instances.
- η is the learning rate of the network.
- $\mathbf{W}^{(k)}[T+1]$ is the weight matrix of the layer k after $T+1$ epochs of training.

3.4. Sentence-level lexical translation scoring

With the independence assumption among target words, the target probabilities are combined to form the sentence-level lexical translation score:

$$p(t|s) = \prod_{t_j \in v_t} p(t_j|s) \tag{3}$$

where v_t is the set of all target words appearing in the target sentence t.

In Equation 3, we need to update the lexical translation score only if a new word appears in the hypothesis. That means we do not take into account the frequency of words

but multiply the probability of one word only once even if the word occurs several times in the sentence. Other models in our translation system, however, will restrict overusing a particular word. Furthermore, to keep track of which words whose probabilities have been calculated already, additional book keeping would be required. In order to avoid those difficulties, we come up with the following approximation given J is the length of the target sentence t:

$$p(t|s) = \prod_{j=1}^{J} p(t_j|s) \quad (4)$$

In order to speed up the calculation of the target word probabilities, we pre-calculate all probabilities for a given source sentence prior to translations. In a naive approach we would need to pre-calculate the probabilities for all possible target words given the source sentence. This would lead to a very slow calculations. Therefore, we first define the target vocabulary of a source sentence as the vocabulary comprised of the respective words from the phrase pairs matching to the source sentence. Using this definition, we only need to pre-calculate the probabilities of all words in the target side of the phrase table and not all target words in the whole corpus. And we can calculate the score for every phrase pair even before starting with the translation.

4. Experiments

In this section, we describe the translation system we use for the experiments, the configurations of the NNDWL and the results of those experiments.

4.1. System description

The system we use as our baseline is a state-of-the-art translation system for English to French without any DWL. To the baseline system, we add several DWL components trained on different corpora as independent features in the log-linear framework utilized by our in-house phrase-based decoder.

The system is trained on the EPPS, NC, Common Crawl, Giga corpora and TED talks[15]. The monolingual data we used to train language models includes the corresponding monolingual parts of those parallel corpora plus News Shuffle and Gigaword. The data is preprocessed and the phrase table is built using the scripts from the Moses package [16]. We adapt the general, big corpora to the in-domain TED data using the Backoff approach described in [17]. Adaptation is also conducted for the monolingual data. We train a 4-gram language model using the SRILM toolkit [18]. In addition, several non-word language models are included to capture the dependencies between source and target words and reduce the impact of data sparsity. We use a bilingual language model as described in [7] as well as a cluster language model based on word classes generated by the MKCLS algorithm [19]. Short-range reordering is performed as a preprocessing step as described in [20].

Our in-house phrase-based decoder is used to search for the best solutions among translation hypotheses and the optimization of the 13 to 17 features, depending on the settings we use, is performed using Minimum Error Rate Training [21]. The weights are optimized and tested on two separate sets of TED talks. The development set consists of 903 sentences containing 20k words. The test set consists of 1686 sentences containing 33k words.

We investigate the impact of our approach by employing different configurations of the neural networks described in details in the following section. We then evaluate those configurations not only for English→French but also for English→Chinese and German→English with similar translation system setups.

Our NNDWL models are trained on a small subset of the mentioned training corpora, mainly the TED data. Although the TED corpus is quite small compared to the overall training data, it is very important since it matches best the test data. In order to speed up the process of testing different configurations, we therefore train the NNDWL only on this corpus except for the comparison reported in Section 4.3.4. The statistics of the training and validation data for the NNDWL are shown in Table 1.

		En-Fr	**En-Zh**	**De-En**
Training	Sent.	149991	140006	130654
	Tok. (avg.)	3.1m	3.3m	2.5m
Validation	Sent.	6153	8962	7430
	Tok. (avg.)	125k	211k	142k

Table 1: *Statistics of the corpora used to train NNDWL*

4.2. Network configurations

In our main neural network architecture we proposed, the sizes of the hidden layers $|H_1|$, $|H_2|$, $|H_3|$ are 1000, 500, 1000, respectively. If we use the original source and target vocabularies, for the English→French direction trained on preprocessed TED 2013 data, V_s includes 47957 words and V_t includes 62660 words. Because of the non-linearity calculations through such a large network, the training is extremely time-consuming. In order to boost the efficiency, we limit the source and target vocabularies to the most frequent ones. All words outside the lists are treated as unknown words. We vary the size of the considered vocabularies from the values $\{500, 1000, 2000, 5000\}$ while keeping the sizes of the hidden layers the same (i.e. $1000 \times 500 \times 1000$). In preliminary experiments, this layout lead to the best performance. So we used this layout for the remaining of the paper.

The same calculation problem occurs with the source contexts, even more seriously due to the curse of dimentionality. Hence, we applied the same cut-off scheme to the source-side bigrams and trigrams with the most-frequent bigram and trigram numbers set at $(200, 100)$, $(500, 200)$ and $(1000, 500)$.

The simpler architecture *SimNNDWL* consisting of one 1000-unit hidden layer is compared to the main architecture with the same setup.

For training our proposed architecture, the gradient descent with a batch size of 15 and a learning rate of 0.02 is used. Gradients are calculated by averaging across a mini-batch of training instances and the process is performed for 35 epochs. After each epoch, the current neural network model is evaluated on a separate validation set, and the model with the best performance on this set is utilized for calculating lexical translation scores afterwards. We regularize the models with the L_2 regularizer. As an alternative to the L_2, we also experiment with the dropout technique [22], where the neurons in the last hidden layer are randomly dropped out with the probability of 0.4. However, it did not help as indicated by its performance on the system later. The training is done on GPUs using the Theano Toolkit[23].

4.3. Results

Here we report the results using different NNDWL configurations mainly for an English→French translation system. We also report the results using the best configurations for other language pairs.

4.3.1. Experiments with different vocabulary sizes

The results of the English→French translation system with NNDWL models trained with different vocabulary sizes are shown in Table 2.

System (En-Fr)	BLEU	ΔBLEU
Baseline	31.94	–
MaxEnt DWL	32.17	+0.23
NNDWL 500	32.06	+0.12
NNDWL 1000	32.37	+0.43
NNDWL 2000	**32.38**	**+0.44**
NNDWL 5000	32.07	+0.13
Full NNDWL	32.06	+0.12

Table 2: *Results of the English→French NNDWL.*

Varying the vocabulary sizes for both source and target sentences not only helps to dramatically reduce neural network training time but also affects the translation quality. In our experiments, neural networks with 1000- and 2000-most-frequent-word vocabularies show the biggest improvements with around 0.44 BLEU points in translating from English to French. They perform better than the DWL using the maximum entropy approach and the NNDWL with the whole source and target vocabularies.

While all NNDWL models achieve notable BLEU gains compared to the strong baseline, some of them are worse than the original MaxEnt model. It might be due to the fact that the original MaxEnt model uses the source contexts whereas the NNDWL models uses just the source words.

4.3.2. The impact of n-gram source contexts

Tables 3 and 4 show the impact of bigrams and trigrams extracted from source sentences. We also vary the numbers of the bigrams and trigrams which appeared most often.

System (En-Fr)	BLEU	ΔBLEU
Baseline	31.94	–
NNDWL 2000	**32.38**	**+0.44**
NNDWL 2000 SC-200-100	32.35	+0.41
NNDWL 2000 SC-500-200	**32.44**	**+0.50**
NNDWL 2000 SC-1000-500	32.36	+0.42

Table 3: *Results of the 2000-NNDWL with source contexts.*

For the NNDWL model with 2000-most-frequent-word vocabularies, including source contexts helps in some cases and does not harm the translation performance in the other cases. With the 500 most-frequent bigrams and 200 most-frequent trigrams, we achieve the best improvements of 0.5 BLEU points over the baseline.

System (En-Fr)	BLEU	ΔBLEU
Baseline	31.94	–
NNDWL 1000	**32.37**	**+0.43**
NNDWL 1000 SC-200-100	32.01	+0.07
NNDWL 1000 SC-500-200	32.23	+0.29
NNDWL 1000 SC-1000-500	**32.39**	**+0.45**

Table 4: *Results of the 1000-NNDWL with source contexts.*

The gains from adding source contexts to the 1000-vocabulary-size NNDWL model are not clearly observed as in the case of the 2000-vocabulary-size model. This might indicate that we should set the numbers of the source contexts to be proportional somehow with the size of the vocabularies.

4.3.3. The impact of using different architectures

System (En-Fr)	BLEU	ΔBLEU
Baseline	31.94	–
NNDWL 1000	**32.37**	**+0.43**
SimNNDWL 1000	32.12	+0.18
NNDWL 2000	**32.38**	**+0.44**
SimNNDWL 2000	32.29	+0.35
NNDWL 5000	**32.07**	**+0.13**
SimNNDWL 5000	31.71	-0.23

Table 5: *Results of NNDWL and SimNNDWL architectures.*

Here we compare our main architecture with the simpler architecture *SimNNDWL* consisting of one 1000-unit hidden layer. While the *SimNNDWL* trains faster (157 hours vs. 202 hours for training English→French with the whole vocabularies), translation time performance is not significantly affected. Since there are decreases in BLEU score using *SimN-*

NDWL architecture as shown in Table 5, the deep architecture seems to have an advantage over the simple architecture. Hence, we stick with our main architecture for remaining experiments.

4.3.4. The impact of data used to train NNDWL models

We also train our NNDWL models on a bigger corpus concatinating EPPS, NC and TED. The results in Table 6 shows that using a bigger corpus does not improve the translation quality. The DWL models trained on in-domain data only, i.e. TED, perform similar or better than the models trained on more data but broader domains. This observation also holds true for original the *MaxEnt DWL* models reported in [24].

System (En-Fr)	BLEU	ΔBLEU
Baseline	31.94	–
NNDWL 1000 on TED	**32.37**	**+0.43**
NNDWL 1000 on EPPS+NC+TED	32.33	+0.39

Table 6: *Results of the NNDWL trained on different corpora.*

4.3.5. Other language pairs

We conducted the experiments with NNDWL models mainly on our English-to-French translation system in order to investigate the impact of our method on a strong baseline. However, we would like to inspect the effect of the DWL on language pairs with long-range dependencies or differences in word order.

For that purpose, we built similar NNDWL models and integrate them to our translation systems for other language pairs. Tables 7 and 8 show the results of English→Chinese and German→English, respectively.

English→Chinese

System (En-Zh)	BLEU	ΔBLEU
Baseline	17.18	–
MaxEnt DWL	16.78	-0.40
NNDWL 500	17.09	-0.09
NNDWL 1000	17.58	+0.40
NNDWL 1000 SC-200-100	**17.63**	**+0.45**
NNDWL 2000	17.26	+0.08
NNDWL 2000 SC-200-100	17.20	+0.02

Table 7: *Results of the English→Chinese NNDWL*

In case of the English→Chinese direction, the NNDWL significantly improves the translation quality, with an increment of 0.45 BLEU points over the baseline. That best BLEU gain comes from the NNDWL with 1000-most-frequent-word vocabularies and the source contexts containing 200 bigrams and 100 trigrams.

German→English

In case of the German→English direction, the NNDWL also helps to gain 0.34 BLEU points over the baseline with the best model (i.e. 2000 most-frequent-word vocabularies with source contexts). However, the improvements is not notably different compared to the original MaxEnt DWL.

System (De-En)	BLEU	ΔBLEU
Baseline	29.70	–
MaxEnt DWL	29.95	+0.25
NNDWL 500	29.82	+0.12
NNDWL 1000	29.92	+0.22
NNDWL 2000	29.95	+0.25
NNDWL 2000 SC-500-200	**30.04**	**+0.34**
NNDWL 5000	29.89	+0.19

Table 8: *Results of the German→English NNDWL*

5. Conclusion

In this paper we described a deep neural network approach for DWL modeling and the integration into a standard phrase-based translation system. Using neural networks as a non-linear classifier for DWL enables the ability of learning the abstract representation of global contexts and their dependencies. We investigated various network configurations on different language pairs. When we deployed our best NNDWL model as a feature in our decoder, it helps to improve up to 0.5 BLEU points compared to a very strong baseline.

Our NNDWL does not require linguistic resources nor feature engineering. Thus, it can easily be ported to new languages. Furthermore, the probability calculation can be done in a preprocessing step. Therefore, the new model would not significantly slow down the translation process. Although we do not feature linguistic resources in our NNDWL, they can be useful in modeling the translation probability of the languages from which they are avalaible. In future work we will try to integrate linguistic features into the model. Moreover, context vector of words might be helpful in further reducing the data sparseness problem.

6. Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

7. References

[1] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[2] P. Koehn, F. J. Och, and D. Marcu, "Statistical Phrase-Based Translation," in *Proceedings of the 2003 Conference of the HLT/NAACL*, 2003.

[3] A. Mauser, S. Hasan, and H. Ney, "Extending Statistical Machine Translation with Discriminative and Trigger-based Lexicon Models," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, ser. EMNLP '09, Singapore, 2009.

[4] J. Niehues and A. Waibel, "An MT Error-driven Discriminative Word Lexicon using Sentence Structure Features," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, 2013, pp. 512–520.

[5] F. J. Och and H. Ney, "The Alignment Template Approach to Statistical Machine Translation," *Computational Linguistics*, vol. 30, no. 4, pp. 417–449, 2004.

[6] J. B. Marino, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. Fonollosa, and M. R. Costa-Jussà, "N-gram-based Machine Translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[7] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, 2011.

[8] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations." in *Proceedings of HLT-NAACL*, 2013, pp. 746–751.

[9] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous Space Translation Models with Neural Networks," in *Proceedings of the 2012 Conference of the NAACL-HLT*, Montréal, Canada, 2012.

[10] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and Robust Neural Network Joint Models for Statistical Machine Translation," in *Proceedings of the Association for Computational Linguistics (ACL), Baltimore*, 2014.

[11] H. Schwenk, "Continuous Space Translation Models for Phrase-based Statistical Machine Translation." in *COLING (Posters)*, 2012, pp. 1071–1080.

[12] S. Bangalore, P. Haffner, and S. Kanthak, "Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction," in *Proceedings of 45th ACL)*, Prague, Czech Republic, 2007, pp. 152–159.

[13] S. Hasan, J. Ganitkevitch, H. Ney, and J. Andrés-Ferrer, "Triplet Lexicon Models for Statistical Machine Translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 372–381.

[14] M. Carpuat and D. Wu, "Improving Statistical Machine Translation Using Word Sense Disambiguation." in *Proceedings of EMNLP-CoNLL*, vol. 7, 2007, pp. 61–72.

[15] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks, year = 2012," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May, pp. 261–268.

[16] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *ACL 2007, Demonstration Session*, Prague, Czech Republic, June 2007.

[17] J. Niehues and A. Waibel, "Detailed Analysis of Different Strategies for Phrase Table Adaptation in SMT," *Proceedings of the Tenth Conference of the Association for Machine Translation in the America (AMTA)*, 2012.

[18] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *Proc. of ICSLP*, Denver, Colorado, USA, 2002.

[19] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes." in *EACL'99*, 1999.

[20] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *TMI*, Skövde, Sweden, 2007.

[21] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, 2003.

[22] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving Neural Networks by Preventing Co-adaptation of Feature Detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[23] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU Math Expression Compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010.

[24] T.-L. Ha, T. Herrmann, J. Niehues, M. Mediani, E. Cho, Y. Zhang, I. Slawik, and A. Waibel, "The KIT Translation Systems for IWSLT 2013," in *Proceedings of the 2013 International Workshop on Spoken Language Translation (IWSLT)*, 2013.

Anticipatory Translation Model Adaptation for Bilingual Conversations

Sanjika Hewavitharana, Dennis Mehay, Sankaranarayanan Ananthakrishnan, Rohit Kumar, John Makhoul

Speech, Language and Multimedia Business Unit
Raytheon BBN Technologies
Cambridge, MA 02138, U.S.A
{shewavit,dmehay,sanantha,rkumar,makhoul}@bbn.com

Abstract

Conversational spoken language translation (CSLT) systems facilitate bilingual conversations in which the two participants speak different languages. Bilingual conversations provide additional contextual information that can be used to improve the underlying machine translation system. In this paper, we describe a novel translation model adaptation method that *anticipates* a participant's response in the target language, based on his counterpart's prior turn in the source language. Our proposed strategy uses the source language utterance to perform cross-language retrieval on a large corpus of bilingual conversations in order to obtain a set of potentially relevant target responses. The responses retrieved are used to bias translation choices towards anticipated responses. On an Iraqi-to-English CSLT task, our method achieves a significant improvement over the baseline system in terms of BLEU, TER and METEOR metrics.

1. Introduction

State of the art conversational spoken language translation (CSLT) systems enable useful, functional communication between two subjects who do not speak the same language. In a typical CSLT pipeline, source language speech is transcribed using automatic speech recognition (ASR), piped to text-to-text statistical machine translation (SMT), followed by text-to-speech (TTS) synthesis in the target language. Two sets of these components are used; one in the source-to-target direction and another in the target-to-source direction. The two directions are typically processed independently, where successive turns in the source and target languages are processed in complete isolation. This decoupling sometimes leads to contextually inappropriate translations.

Fortunately, bilingual conversations offer a wealth of contextual information that can be exploited to improve translation performance. Contextual cues can be used to adapt the translation model and improve its relevance to the current state of the dialogue. Typically, the adaptation is done monolingually, using only the utterances of one speaker. In this paper, we describe a novel translation model adaptation technique for bilingual conversations that *anticipates* a participant's response in the target language based on his *counterpart's* prior turn in the source language. Depending on the nature of the bilingual conversation, adaptation can be profitably performed in either language. We evaluate the proposed approach on Iraqi-English bilingual conversations drawn from the DARPA TransTac/BOLT spoken dialogue collection.

Our approach is motivated by the observation that in most domains, the primary goal of bilingual conversations is to exchange information across the language barrier. To that end, the most useful translation is often the one that most effectively conveys the content of a speaker's *response* to the content of the *counterpart's preceding utterances*. Table 1 illustrates this with an excerpt from an Iraqi-English bilingual conversation at a vehicle checkpoint from the DARPA TransTac/BOLT corpus. The first column corresponds to the English speaker's turn; the second column is the Iraqi speaker's following turn, or response (in Buckwalter transliteration); the third column provides an English gloss of the Iraqi speaker's response. As in most cooperative conversations, the Iraqi responses are all relevant to the preceding English turn, and, in many cases, largely predictable from the preceding English turn in the first column.

Following these observations, we perform turn-level translation model adaptation that prefers phrasal translation rules that originate from *responses* that immediately follow *counterpart utterances* that are similar to those of the *current conversational counterpart*. This approach produces a measurable improvement over a phrase-based SMT baseline system in terms of BLEU, TER and METEOR metrics on an Iraqi-to-English translation task.

2. Anticipatory Translation Model Adaptation

Our adaptation scheme attempts to model the effect of the preceding target language turn on the translation of the current source language utterance. The intuition is that biasing

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.
Approved for Public Release, Distribution Unlimited

N	English Turn	Iraqi Response	English Gloss of Iraqi Response
1.	*turn off your engine and get out of the car*	*tfDlwA*	*here you are*
2.	*give me your i_d*	*bTAqty wjwAzy*	*my i_d card and my passport*
3.	*where you coming from*	*mn swryA mn dyr Alzwr*	*from syria from dair al-zour*
4.	*and where you going*	*rAyH llrmAdy*	*i'm going to ramadi*
5.	*what's in your truck*	*Iny bqAl wdJjyb xs JbyEh hnAk*	*i'm a grocer and i'm bringing lettuce*

Table 1: Excerpt from an Iraqi-English bilingual conversation in the DARPA TransTac/BOLT collection.

the translation model to favor phrase pairs originating from training utterances that have similar preceding target language turn will produce translations more appropriate to the current conversation. Such a model can be learned in a data-driven fashion from a large training corpus of bilingual conversations, organized in the form of starting target language turns and ensuing source language responses. The DARPA TransTac/BOLT spoken dialogue corpus is organized as a collection of bilingual conversations, thus making it relatively simple to build an "anticipatory parallel corpus" (APC) of target language turn and source language response pairs for training the translation model (see Section 3). The APC is a pseudo-parallel corpus with prior target turns mapping to the immediately following source language responses, similar to the first two columns of Table 1. In the following description, we assume, without loss of generality, that we are performing cross-lingual translation model adaptation for translating the current Iraqi turn into English based on the preceding English turn. Figure 1 illustrates the adaptation process.

2.1. Cross-Lingual Retrieval

When decoding the current Iraqi utterance in the context of a bilingual conversation, we seek to predict what an appropriate response to the preceding English turn might look like.[1] To find support for this prediction, we use the preceding English turn as a query to perform cross-lingual retrieval on the APC constructed from the training conversations. The goal of this step is to obtain the most relevant Iraqi responses to the preceding English turn. Each training utterance pair in the APC is assigned a unique utterance ID, which we later use in the online adaptation of the translation model (Section 2.2).

Because the APC is not a true parallel corpus in the sense that the Iraqi responses are not direct translations of the preceding English turns, learning a true cross-lingual retrieval model from this data would be difficult. Instead, we employed the simpler approach of first performing *monolingual* retrieval of the English turns most similar to the query turn, and then reading off the corresponding Iraqi responses from the APC. To facilitate this, we represent all APC English turns in a trigram term-indicator vector space with appropriate pre-processing (e.g. stop-word removal), and we index each training utterance separately. During retrieval, we map the preceding English turn to the same vector space, and select APC English turns that have the largest cosine similarity to the query. We then read off the corresponding Iraqi response turns from the APC. This produces a *bias corpus* of Iraqi responses that might be relevant to the preceding English turn, which we limit to a small number between 50 and 500 in our experiments.

Table 2 illustrates anticipatory cross-lingual retrieval with an example. The first row corresponds to the query English turn. The first column of the second row lists the five top-ranking Iraqi responses retrieved from the APC using the above mechanism. The second column of the second row provides an English gloss for the retrieved Iraqi responses. The final row shows the actual Iraqi response to the query English turn, and its English gloss. In this example, the retrieved Iraqi responses are well-matched to the actual response. Thus, a translation model biased towards the phrases extracted from the retrieved responses is likely to produce better translations.

Q.	**how are you doing today**	
1.	*wAllh AlHmd llh zyn*	*well fine thank god*
2.	*SbAH Alnwr JhlAF wshlAF*	*good morning hello and welcome*
3.	*JhlAF byk kyf AlHAl*	*hello to you how are you*
4.	*SbAH Alxyr JhlAF wshlAF AlHmd llh zyn*	*good morning and welcome thank god i'm well*
5.	*Iny zyn JHsn mn Endh*	*i'm fine better than him*
R.	**AlHmd llh zyn**	**good thank god**

Table 2: Iraqi response retrieval for a sample English query turn.

2.2. Translation Model Adaptation

From the cross-lingual retrieval on each previous English turn, we obtain for each Iraqi turn I, a set of anticipated Iraqi responses, corresponding Iraqi utterance IDs and a set of similarity scores (cosine similarity between the query English turn and APC English turns) $\mathbf{R}$. We use these scores directly as relevance scores for the anticipatory Iraqi responses. At run time, an updated relevance vector is passed on to the

[1] In an interactive CSLT system, all utterances by speakers of both languages are transcribed by speech recognition (ASR), though we also run oracle experiments with the ground truth (reference) transcription.

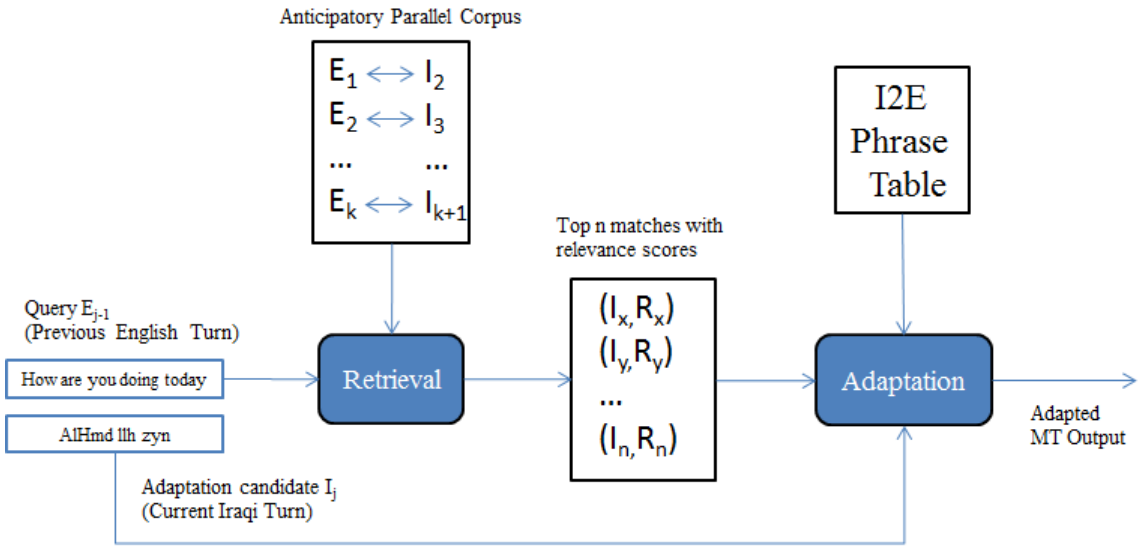


Figure 1: Anticipatory translation model adaptation process.

SMT decoder for each new test utterance.

The SMT phrase table tracks, for each phrase pair, the set of training utterances from which that phrase pair originated. Only part of the training corpus has marked conversation boundaries. Phrase translation rules derived from sentence pairs that do not originate in bilingual conversations are assigned a default utterance ID. For each candidate phrase pair $\overline{I} \rightarrow \overline{E}$ added to the search graph, the SMT decoder computes the relevance score as the maximum of all relevance scores corresponding to the current turn. i.e.

$$F_{\overline{I}\rightarrow\overline{E}} = \max_{j \in Par(\overline{I}\rightarrow\overline{E})} \mathbf{R}_j \quad (1)$$

where $Par(\overline{I} \rightarrow \overline{E})$ is the set of training utterances from which the candidate phrase pair originated. Phrase pairs with the default utterance ID are assigned a default relevance score of 0.0. (in effect, they are decoded with the baseline features only). The relevance score is added as a feature to the log-linear translation model with its own weight, which is tuned with the rest of the parameters. The effect of this feature is to bias the decoder in favor of phrase pairs that originate in relevant responses.

3. Baseline SMT System

We use the DARPA TransTac/BOLT Iraqi-English parallel two-way spoken dialogue collection to train the translation models. Each conversation represents an interaction between an English interviewer and an Iraqi respondent, based on a scenario that requires exchange of specific information. The English speaker typically plays the role of information seeker and "drives" the majority of conversations. These large-vocabulary conversations are spontaneous and free-form, with few restrictions. This collection consists of a variety of domains including force protection (e.g. checkpoint, reconnaissance, patrol), medical diagnosis and aid, maintenance and infrastructure, etc; each transcribed from spoken bilingual conversations and manually translated. The SMT parallel training corpus contains approximately 773K sentence pairs (7.3M English words). We used this corpus to extract translation phrase pairs from bidirectional IBM Model 4 word alignment [1] based on the heuristic approach of [2]. A 4-gram target LM was trained on all English transcriptions. Our phrase-based decoder is similar to Moses [3] and uses the phrase pairs and target LM to perform beam search stack decoding based on a standard log-linear model, the parameters of which were tuned with MERT [4] on a held-out development set (≈11,000 sentence pairs) using BLEU as the tuning metric. Finally, we evaluated translation performance on a separate, unseen test set (≈9,300 sentence pairs). Most of these conversations between bilingual speakers are mediated through a human interpreter.

Of the 773K training sentence pairs, about 267K originate in ≈3,000 marked-up bilingual conversations. We use this subset to construct an anticipatory corpus for the adaptation experiments. These sentence pairs are assigned a unique utterance ID. All other sentence pairs are assigned to a default utterance ID, which signals the absence of the anticipatory relevance feature for phrase pairs derived from these instances.

4. Experimental Results

We constructed an English-Iraqi APCs from input-response pairs in the training conversations. For each source language input turn in the held-out development and test sets, we performed cross-lingual retrieval on the APC to obtain a bias corpus of potential responses in the target language. We performed retrieval in two configurations: (a) using reference transcriptions of all utterances in both languages; and (b)

using ASR transcriptions (both for retrieval and translation) in both languages. The latter configuration degrades performance noticeably, but it matches the conditions of a live deployment. In the Iraqi-English experiments, we test values of the relevance list size $n \in \{50, 100, 500\}$.

The Iraqi ASR transcriptions were generated using a two-pass HMM-based system, which delivered a word error rate (WER) of 20.2% on the test set utterances. The English ASR system, which was used to transcribe the counterpart's utterances had a WER of 10.6%.

The held-out development conversations were used to tune the size of the bias corpus (i.e the number of retrieved response turns), as well as the model weights in the log-linear translation model. Tuning was performed using reference transcriptions of the Iraqi turn. The optimal settings were then used to decode the unseen test conversations for both reference transcriptions and ASR transcriptions.

REFERENCE TRANSCRIPTIONS			
SYSTEM	BLEU↑	TER↓	METEOR↑
Baseline	31.62	53.32	63.59
n=50	31.73*	53.11*	63.67
n=100	**31.82***	53.03*	**63.75**
n=500	31.80*	**53.00***	**63.75**
ASR TRANSCRIPTIONS			
SYSTEM	BLEU↑	TER↓	METEOR↑
Baseline	26.93	60.38	58.20
n=50	26.98	60.12	58.21
n=100	**27.11***	60.16*	**58.26**
n=500	27.01	**60.06***	58.25

Table 3: Translation results on the test sets. Asterisked results are significantly better than the baseline ($p \leq 0.05$) using 1,000 iterations of paired bootstrap re-sampling [5]. Best results for each metric are marked in boldface.

Table 3 summarizes the translation performance of the test sets in BLEU [6], TER [7] and METEOR [8]. Results are presented for three configurations of n: 50, 100 and 500. We note that our proposed anticipatory adaptation approach outperforms the baseline across multiple metrics, both reference transcriptions and ASR transcriptions. In many instances, the differences are statistically significant. The adapted system with 100 retrieval responses (n=100) is the best scoring system for that test set.

In Table 4 we show example utterances where our adaptation approach generates better translation choices. In these examples, the conversational counterpart's utterance guides the retrieval towards contextually relevant matches, which influence lexical (hence, phrasal) selection (e.g. 'flight of stairs' vs. 'stairs' in a conversation about a corridor). Retrieval-based adaptation can also go awry, as the fourth example shows. In this example, the brevity of the preceding English turn leads to imprecise retrieval and an unreliable bias corpus, which then prefers an incorrect translation for incidental reasons.

We also compared smoothed, sentence-level BLEU scores,[2] and observed that the the n=100 adapted system scores higher than the baseline 884 times and lower than the baseline 763 times.[3] We take this as further evidence that the retrieval-based adaptation leads to small but systematic improvements in translation quality.

5. Relation to Prior Work

Online model adaptation for SMT has become an active area of research in recent years. The predominant approach is to divide the training data into discrete partitions representing either *domains* or *genres* to be adapted to [9, 10] or other linguistic phenomena of interest, such as whether the current utterance is a *question* [11]. At run-time, the domain, genre or other inferred properties of the current utterance are used to prefer phrase translation rules that originate in appropriate training data. By contrast, our approach makes no assumptions about the nature of the training data, and therefore requires no hard decisions about training set partitions and no labor-intensive manual annotation. Instead, we directly retrieve exemplars from the training set using lexical cues in order to guide the anticipatory inference.

To avoid the need for hard decisions about domain membership, some have used topic modeling to improve SMT performance, e.g., using latent semantic analysis [12], 'biTAM' [13] or latent dirichlet allocation [14, 15, 16]. As it also avoids data set partitioning and explicit annotation, our work is in the same spirit as these, but we do not explicitly model topic distributions.

In our previous work [16], we *incrementally* accumulated conversational history to compute a topic distribution vector. The phrasal translation rules were scored using the maximum similarity of the current conversational topic vector to all of the training conversation topic vectors from which that phrasal rule was drawn. This work is also incremental, but in contrast uses only the previous utterance of the conversational counterpart to retrieve exemplars for similarity comparisons. Here, we score phrasal rules using the maximum similarity of all of the retrieved sentences to any of the sentences from which the phrase pair was drawn.

6. Discussion and Future Directions

Conversational spoken language translation systems offer rich contextual cues that can be used to improve the MT performance. This in turn results in more usable, higher quality CSLT systems that are better able to accomplish cross-lingual communication goals in a way that is tailored to the conversation at hand. In this paper, we described a novel, turn-level anticipatory translation model adaptation technique where one participant's turn is used to anticipate,

[2] As computed by the NIST BLEU script.

[3] Of the remaining 7,662 utterances, the two systems differ in their translations of 1,867, even though their BLEU scores do not differ.

Previous Eng Turn	but his temperature how has he been hotter than normal
Baseline	*his temperature sometimes and his body is very hot*
Adaptive	*his temperature goes up sometimes and his body is very hot*
Reference	his temperature sometimes goes up and his body becomes very hot
Previous Eng Turn	can you see this corridor in front of you
Baseline	*this is the end there are stairs*
Adaptive	*at the end of it there is a flight of stairs*
Reference	at the end of it there's a staircase
Previous Eng Turn	if you can't stop it then that is an emergency situation
Baseline	*of course they call it the pressure the direct pressure on the wound or continuous*
Adaptive	*of course they call it direct pressure or continuous pressure on the wound*
Reference	of course they call it direct pressure or continuous pressure on the wound
Previous Eng Turn	good
Baseline	*personally because he is supposed to*
Adaptive	*personally because he is the foundation*
Reference	to him personally because he is the one concerned [...with the matter]

Table 4: Examples of Iraqi-to-English translations where anticipatory adaptation influences the lexical choice.

and thereby more accurately translate, the other participant's response.

The proposed approach used cross-lingual retrieval on an "anticipatory parallel corpus" of target language turns and corresponding source language responses to obtain the most relevant responses to a query turn. The retrieved responses were used to bias translation options in the translation model for the subsequent response turn in an Iraqi-Arabic-to-English translation system. We observed statistically significant improvements in translation results for most of the testing conditions, which included both reference and ASR transcripts of the bilingual test conversations. We also showed examples where the proposed approach produced better translations than the baseline system.

In this paper, we demonstrated the usefulness of turn-level context of bilingual conversations for improving MT performance. Our next goal is to develop a framework for integration of fine-grained turn-level translation model adaptation with more coarse-grained, globally driven approaches such as topic-based translation model adaptation, possibly in a neural-network-based translation model (such as [17]) where diverse sources of information can be combined to make more informed translation choices. We also plan to explore ways of detecting unreliable retrieval query input (e.g., short preceding conversational turns, as in Table 4) that can lead to unreliable translation biasing.

7. Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful feedback. This work was funded in part by the DARPA BOLT program under contract number HR0011-12-C-0014.

8. References

[1] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, March 2003. [Online]. Available: http://dx.doi.org/10.1162/089120103321337421

[2] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *NAACL-2003: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 48–54.

[3] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL-2007. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 177–180. [Online]. Available: http://dl.acm.org/citation.cfm?id=1557769.1557821

[4] F. J. Och, "Minimum error rate training in statistical machine translation," in *ACL-2003: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 160–167.

[5] P. Koehn, "Statistical significance tests for machine translation evaluation," in *EMNLP*, Barcelona, Spain, July 2004, pp. 388–395.

[6] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *ACL-2002: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2001, pp. 311–318.

[7] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings AMTA*, August 2006, pp. 223–231. [Online]. Available: http://www.mt-archive.info/AMTA-2006-Snover.pdf

[8] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72. [Online]. Available: http://www.aclweb.org/anthology/W/W05/W05-0909

[9] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," in *Proceedings of the Second Workshop on Statistical Machine Translation*, ser. StatMT-2007. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 128–135. [Online]. Available: http://dl.acm.org/citation.cfm?id=1626355.1626372

[10] S. Matsoukas, A.-V. I. Rosti, and B. Zhang, "Discriminative corpus weight estimation for machine translation," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, ser. EMNLP-2009. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 708–717. [Online]. Available: http://dl.acm.org/citation.cfm?id=1699571.1699605

[11] A. Finch and E. Sumita, "Dynamic model interpolation for statistical machine translation," in *Proceedings of the Third Workshop on Statistical Machine Translation*, ser. StatMT-2008. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 208–215. [Online]. Available: http://dl.acm.org/citation.cfm?id=1626394.1626428

[12] Y.-C. Tam, I. Lane, and T. Schultz, "Bilingual LSA-based adaptation for statistical machine translation," *Machine Translation*, vol. 21, no. 4, pp. 187–207, Dec. 2007. [Online]. Available: http://dx.doi.org/10.1007/s10590-008-9045-2

[13] B. Zhao and E. P. Xing, "BiTAM: Bilingual topic admixture models for word alignment," in *In Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL '06)*, 2006.

[14] Z. Gong, Y. Zhang, and G. Zhou, "Statistical machine translation based on LDA," in *Universal Communication Symposium (IUCS), 2010 4th International*, 2010, pp. 286–290.

[15] V. Eidelman, J. Boyd-Graber, and P. Resnik, "Topic models for dynamic translation model adaptation," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ser. ACL-2012. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 115–119. [Online]. Available: http://dl.acm.org/citation.cfm?id=2390665.2390694

[16] S. Hewavitharana, D. N. Mehay, S. Ananthakrishnan, and P. Natarajan, "Incremental Topic-Based Translation Model Adaptation for Conversational Spoken Language Translation." in *ACL-2013: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 697–701.

[17] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2014, pp. 1370–1380.

Offline Extraction of Overlapping Phrases for Hierarchical Phrase-Based Translation

Sariya Karimova, Patrick Simianer, Stefan Riezler

Computational Linguistics, Heidelberg University
69120 Heidelberg, Germany
{karimova,simianer,riezler}@cl.uni-heidelberg.de

Abstract

Standard SMT decoders operate by translating disjoint spans of input words, thus discarding information in form of overlapping phrases that is present at phrase extraction time. The use of overlapping phrases in translation may enhance fluency in positions that would otherwise be phrase boundaries, they may provide additional statistical support for long and rare phrases, and they may generate new phrases that have never been seen in the training data. We show how to extract overlapping phrases offline for hierarchical phrase-based SMT, and how to extract features and tune weights for the new phrases. We find gains of $0.3 - 0.6$ BLEU points over discriminatively trained hierarchical phrase-based SMT systems on two datasets for German-to-English translation.

1. Introduction

Decoding in SMT amounts to searching for the most probable (Viterbi) derivation of a target string given the source string. Standard SMT decoders perform at the same time a search for the optimal segmentation of the source sentence into disjoint spans of words, which are translated by rules encoding bi-phrases. This means that irrespective of whether phrases are contiguous [1], non-contiguous [2, 3], or hierarchical [4], the application of phrase rules at decoding time disallows overlapping words. However, the use of overlapping phrases might have several advantages: First, they may enhance fluency in positions that would otherwise be phrase boundaries. Second, overlapping phrases may provide additional statistical support for long and rare phrases extracted from the training data. Finally, and most importantly, overlapping phrases may constitute new phrases that have never been seen in the training data but may be applicable to the test data.

The few approaches that did attempt to integrate overlapping phrases into SMT decoding in the past [5, 6, 7] were handicapped mostly by the additional decoding complexity. The need to counterbalance exponential growth of the search space with very restrictive reordering constraints prevented these approaches to be competitive with state-of-the-art phrase-based SMT. The exception is Tribble et al. [8] who reported significant gains for using overlapping phrases over their own baseline. The key idea in this approach is to circumvent decoder integration and instead to generate overlapping phrases *offline*, by merging existing contiguous phrases into longer bi-phrases that have overlapping words in both source and target.

In this work, we will revive this approach, and extend it to hierarchical phrases. We show how to merge and filter overlapping phrases created from hierarchical and non-hierarchical phrases, and how to extract and tune features for the new phrases. An experimental comparison with a state-of-the-art hierarchical phrase-based decoder [9] shows gains of $0.3 - 0.6$ BLEU points on two datasets for German-to-English translation.

2. Related Work

The potential of overlapping phrases to improve fluency and to smooth prediction of long and rare phrases has been discovered independently in a few instances in prior work. The crux of most of these approaches is an efficient integration of overlapping phrases into decoding. For example, the exponential number of translation hypotheses arising from overlapping phrases has been managed in beam search decoding frameworks by reordering constraints that allow only adjacent non-overlapping phrases to be swapped [5, 7]. This reordering constraint seems to be too restrictive since it impacts translation quality in comparison to state-of-the-art phrase-based SMT.

Alternatively, sampling-based approaches [6] or graph-search techniques [10, 11, 12] have been used for decoding with overlapping phrases. These approaches suffer from search errors due to necessary abstractions in sampling or due to necessary approximations in adaptation of graph search algorithms to SMT decoding.

The work related closest to our approach is that of Tribble et al. [8, 13]. Their key idea is to circumvent decoder integration and instead to generate overlapping phrases in an offline manner. In contrast to our work, their approach is restricted to merging contiguous phrases. Furthermore, they extract a single feature (based on phrase-internal word alignments) for new phrases and do not learn discriminative weights. A similar idea has also been presented for Example-Based MT

[14, 15] where the focus is on combining given overlapping phrases by a new search algorithm.

An alternative to enriching the repository of phrases with overlapping phrase rules is the design of context-sensitive features for discriminative training. Target context is clearly exploited by large language models. Word-sense disambiguation inspired features [16] allow to exploit source context, and recent approaches successfully merged source and target context into a powerful decoding feature [17]. However, these approaches are orthogonal to our work.

3. Generating Overlapping Phrases with and without Variables

Hierarchical phrases can be formalized as rules of a synchronous CFG [4]. We denote terminals consisting of contiguous phrases by T, and the single non-terminal variable by NT. The key idea is to merge base rules into new rules by pivoting on overlapping words. We apply this idea to base rules consisting of terminals only (T rules) and to base rules including non-terminals (NT rules).

As a first step, we apply the technique of [18] to extract rules for German-to-English translation from the News Commentary and TED data (see Section 5.1). Tables 1 and 2 show the token counts of rule shapes for the extracted grammars.

We see that base rules consisting of terminals only (rule shape T-T) are quite frequent in the extracted grammars for both datasets. To these rules, the ideas of [8], namely merging all base rules that have overlapping words on both source and target can be applied directly. For base rules including non-terminals (rule shape including NT), merging of rules can be done at word overlaps in terminals at the head of one rule with terminals at the tail of another rule.

Because of the huge number of potential new rules, we apply several filtering steps to the merging process. For T rules, we firstly restrict our attention to base rules with more than one terminal on source and target side. Secondly, we apply count cutoffs of less than 5, 8, and 11 occurrences of base rules in the training set. Lastly, given the test set, we only store merged rules whose source sides are in principle applicable to the test set. For rules including NTs, we restrict our attention to base rules with exactly one NT on source and target. Furthermore, we consider only base rules that are seen at least 17, 20, or 23 times in the training set. Lastly, a pre-filtering based on applicability of merged rules to test set sources is done. Tables 3 and 4 show the counts of base rules and merged rules before and after filtering on the News Commentary and TED datasets.

Overall, these filtering steps resulted in a considerable number of new rules, i.e., rules that are unseen in the training set. Table 5 shows the percentages of overlapping phrase rules that are applicable to the test data, but are unseen in the training data, together with their actual use in the 1-best translation of the test data. We find that new rules are composed at a considerably higher percentage from base T rules than from base NT rules, resulting in a similar usage pattern

	News Commentary testset		TED testset	
	new	used	new	used
T 5	65.3	25.3	63.5	54.5
T 8	54.8	18.7	49.6	43.3
T 11	47.1	10.6	40.1	36.7
NT 17	21.7	2.5	37.8	10.8
NT 20	17.7	4.5	35.2	8.6
NT 23	15.3	5.5	32.7	6.9
T + NT	24.7	16.4	38.03	23.0

Table 5: Percentages of overlapping phrase rules composed from base rules and unseen in training ("new"), out of rules of the same form applicable to the test set, together with their usage in translating the test set ("used"), out of rules of the same form used to translate the test set.

(1) $X \rightarrow$ ⟨ es stellt sich heraus ||| it turns out ⟩

(2) $X \rightarrow$ ⟨ stellt sich heraus , dass ||| turns out that ⟩

(3) $X \rightarrow$ ⟨ es stellt sich heraus , dass ||| it turns out that ⟩

Figure 1: T rule (3) merged from rules (1) and (2).

of more T rules than NT rules used in 1-best translations of both test sets. As expected, these percentages are decreasing the more restrictive the count cutoffs are set. A combination of T and NT rules shows a pattern of composition and usage in between T rules and NT rules.

Across all extracted rules, the average number of words in merged rules is as little as 0.1 tokens higher than in base rules for News Commentary, and increases on average up to more than 1 token for the TED data set. For the majority of cases, the overlap is 1 token in source and target. In $1 - 2\%$ of the cases, the overlap is 2 tokens, and only 0.1% of the new phrases overlap in 3 or 4 tokens.

An example for a merger of two T rules (1) and (2) into a new rule (3), with an overlap of 3 source tokens and 2 target tokens, is given in Fig. 1. A merger of two rules including NTs is given in Fig. 2. Here, the overlap in target and source is 2 tokens.

4. Feature Extraction and Tuning

[8] use IBM model 1 word-level alignments of the merged phrases to directly assign probabilities to the new phrases. In this work, we use the SMT decoder `cdec` [9] that combines features into a log-linear model and offers several learners for discriminative tuning of weights.

We compare four feature configurations. First, we use

Count	Shape
359,406	T NT T - T NT T
270,813	NT T NT T - NT T NT T
267,528	T NT T NT - T NT T NT
155,250	T NT T NT T - T NT T NT T
129,400	T - T
109,447	T NT - T NT
104,924	NT T - NT T
99,615	NT T NT - NT T NT
58,824	T NT T NT - T NT NT
50,253	NT T NT T - NT NT T
35,015	T NT T NT T - T NT NT T
28,496	NT T NT T - T NT NT T
24,523	T NT T - T NT
23,821	T NT T NT - T NT NT T
22,705	NT T - T NT
20,658	NT T NT - T NT NT
20,639	NT T NT T - NT T NT
20,498	NT T NT - NT NT T
20,455	T NT - NT T

Count	Shape
20,003	NT T - T NT T
17,480	NT T NT - T NT T NT
17,276	T NT T - NT T
16,967	NT T NT T - T NT T NT T
16,559	T NT T NT - NT T NT
16,465	T NT T NT - T NT T NT T
15,965	NT T NT - NT T NT T
15,366	T NT - T NT T
11,736	T NT T NT T - T NT T NT
11,378	T NT T NT T - NT T NT T
10,625	NT T NT T - T NT T NT
8,691	T NT T NT - NT T NT T
2,693	NT T NT T - T NT NT
1,948	NT T NT - T NT NT T
1,525	T NT T NT - NT NT T
848	NT T NT - T NT T NT T
576	T NT T NT T - NT T NT
459	T NT T NT T - T NT NT
303	T NT T NT T - NT NT T

Table 1: Rule shapes in the grammar extracted from News Commentary.

(1) $X \rightarrow \langle$ ist wirklich X_1 , aber ||| is really X_1 , but $\rangle$

(2) $X \rightarrow \langle$, aber man X_1||| , but you $X_1 \rangle$

(3) $X \rightarrow \langle$ ist wirklich X_1 , aber man X_2|||
is really X_1 , but you $X_2 \rangle$

Figure 2: NT rule (3) merged from rules (1) and (2).

`cdec`'s implementation of lexical phrase probabilities for source words f and target words e:

$$MaxLexFgivenE = -\sum_i \log_{10} p_{max}(f_i|e) \quad (1)$$

and

$$MaxLexEgivenF = -\sum_i \log_{10} p_{max}(e_i|f). \quad (2)$$

Second, we add a new feature that indicates whether a rule is created by merging as follows:

$$NewRule = \begin{cases} 1 & \text{if the rule is new,} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Third, we calculate the following standard statistics among new rules that were merged from base rules extracted for the test set:

$$EgivenFCoherent = -\log_{10}(count\_EF/count\_F) \quad (4)$$

$$SampleCountF = \log_{10}(1 + count\_F) \quad (5)$$

$$CountEF = \log_{10}(1 + count\_EF) \quad (6)$$

$$IsSingletonF = \begin{cases} 1 & \text{if } count\_F = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

$$IsSingletonFE = \begin{cases} 1 & \text{if } count\_EF = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Last, we take inspiration from [19]'s adaptive features that combine counts from a lookup in post-editing data with counts from the suffix array sample extracted for the test set. In our case, this corresponds to combining count statistics for new rules only (denoted by subscript $\mathcal{L}$) with count statistics for base rules extracted for the test set (denoted by subscript $\mathcal{S}$):

$$EgivenFCoherent = -\log_{10}((count\_EF_{\mathcal{S}} + count\_EF_{\mathcal{L}})/ (count\_F_{\mathcal{S}} + count\_F_{\mathcal{L}})) \quad (9)$$

$$SampleCountF = \log_{10}(1 + count\_F_{\mathcal{S}} + count\_F_{\mathcal{L}}) \quad (10)$$

Count	Shape
373,500	T NT T - T NT T
284,364	T NT T NT - T NT T NT
277,682	NT T NT T - NT T NT T
204,562	T NT T NT T - T NT T NT T
97,485	T - T
92,133	T NT - T NT
86,469	NT T - NT T
85,518	NT T NT - NT T NT
47,617	T NT T NT - T NT NT
43,403	T NT T NT T - T NT NT T
38,121	NT T NT T - NT NT T
29,213	NT T NT T - T NT NT T
25,302	T NT T NT - T NT NT T
20,839	T NT T - T NT
20,173	NT T NT T - T NT T NT T
17,559	NT T NT T - NT T NT
17,328	NT T - T NT T
16,404	NT T NT - T NT T NT
16,087	T NT T NT - T NT T NT T

Count	Shape
14,166	T NT T NT T - T NT T NT
14,039	NT T NT - NT T NT T
13,836	T NT T - NT T
13,476	T NT - T NT T
13,078	NT T NT - NT NT T
12,907	T NT T NT - NT T NT
12,893	NT T - T NT
12,658	T NT - NT T
12,376	NT T NT - T NT NT
10,454	T NT T NT T - NT T NT T
7,159	NT T NT T - T NT T NT
5,170	T NT T NT - NT T NT T
1,566	NT T NT - T NT NT T
1,368	NT T NT T - T NT NT
836	T NT T NT - NT NT T
813	NT T NT - T NT T NT T
496	T NT T NT T - NT T NT
343	T NT T NT T - T NT NT
234	T NT T NT T - NT NT T

Table 2: Rule shapes in the grammar extracted from TED talks.

$$CountEF = \log_{10}(1 + count\_EF_{\mathcal{S}} + count\_EF_{\mathcal{L}}) \quad (11)$$

$$MaxLexFgivenE = p_{max}(\tilde{f}|\tilde{e}) = -\sum_i \log_{10} p_{max}(f_i|e) \quad (12)$$

$$MaxLexEgivenF = p_{max}(\tilde{e}|\tilde{f}) = -\sum_i \log_{10} p_{max}(e_i|f) \quad (13)$$

$$IsSingletonF = \begin{cases} 1 & \text{if } count\_F_{\mathcal{S}} + count\_F_{\mathcal{L}} = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

$$IsSingletonFE = \begin{cases} 1 & \text{if } count\_EF_{\mathcal{S}} + count\_EF_{\mathcal{L}} = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

$$NewRule = \begin{cases} 1 & \text{if the rule is new,} \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

Discriminative tuning is performed on the respective tuning sets of the News Commentary and TED data. We use the pairwise ranking learner of [20] for this purpose. In addition to the standard handful of dense feature, sparse features for rule shapes, rule identifiers, and bigrams in rule source and target are extracted from grammar rules.

NC	train	train-lm	tune	test
Sentences	136,227	180,657	1,057	1,064
Words de	3,005,252		26,205	23,593
Words en	2,909,346	3,797,500	25,660	22,518
TED	train	train-lm	tune	test
Sentences	139,563	158,641	1,172	746
Words de	2,195,030		21,270	11,831
Words en	2,332,370	2,715,777	21,679	12,734

Table 6: News Commentary and TED de-en parallel data.

5. Translation Experiments

5.1. Systems and Data

The data used in our experiments are the German-English parallel data provided in the News Commentary and TED releases of WMT 2007[1] and IWSLT 2013[2], respectively. Table 6 gives the basic data statistics for News Commentary (NC) and TED data.

The bilingual SMT system used in our experiments is the state-of-the-art SCFG decoder `cdec` [9][3]. We built grammars using its implementation of the suffix array extraction method described in [18]. Word alignments are built from all parallel data using fast_align [21]. SCFG models use the same settings as described in [4]. For language modeling, we built a modified Kneser-Ney smoothed 5-gram language

[1] `http://statmt.org/wmt07/shared-task.html`
[2] `http://www.iwslt2013.org/`
[3] `http://www.cdec-decoder.org`

News Commentary	Base rules	Merged rules	Unique	Applicable in test	Unique
all	129,400				
>1 token	72,322				
T 5	6,823	364,642	352,171	6,311	5,739
T 8	4,434	171,715	167,125	3,414	3,165
T 11	3,286	100,513	98,268	2,203	2,054
TED	Base rules	Merged rules	Unique	Applicable in test	Unique
all	97,485				
>1 token	62,671				
T 5	6,073	370,611	363,789	8,823	7,637
T 8	4,088	181,227	178,010	4,828	4,235
T 11	3,115	105,657	103,906	3,203	2,855

Table 3: Counts of base rules and merged rules with terminals only before and after filtering.

News Commentary	Base rules	Merged rules	Unique	Applicable in test	Unique
all	694,105				
NT 17	14,107	563,980	556,476	18,588	14,919
NT 20	11,592	324,790	319,919	13,794	11,039
NT 23	9,774	198,447	194,880	10,915	8,690
TED	Base rules	Merged rules	Unique	Applicable in test	Unique
all	643,132				
NT 17	14,684	1,980,618	1,940,402	34,696	28,293
NT 20	12,256	1,345,298	1,316,680	26,856	21,750
NT 23	10,334	908,066	887,474	21,118	16,938

Table 4: Counts of base rules and merged rules with nonterminals before and after filtering.

model [22, 23].

All data were normalized, tokenized and lowercased; German compounds were split. For tokenization, lowercasing and other preprocessing steps we used the scripts distributed with the Moses SMT toolkit [24]. For compound splitting in German texts a standard empirical approach of [25] was employed.

5.2. Experimental Results

Table 7 shows BLEU [26] results for MERT [27] optimization of dense feature weights, and for pairwise ranking [20] optimization of sparse feature weights. MERT runs were repeated three times to account for optimizer instability [28]. The pairwise ranking technique was stable in this respect. Statistical significance is measured using Approximate Randomization [29, 30] where result differences with a p-value smaller than 0.05 are considered significant.

In order to investigate a possible correspondence of the patterns of composition and usage shown in Table 5, we evaluate overlapping phrases merged from base T rules and base NT rules separately. Table 8 shows BLEU results for different frequency cutoffs for base rules (see Section 3) and different feature sets (see Section 4) on the News Commentary data for German-to-English translation. All results are nominal improvements over the PairRank baseline in Table 7, with several statistically significant result differences. Best results, namely an improvement of 1.3 BLEU points over the MERT baseline, and a gain of 0.6 BLEU points over the pairwise-ranking baseline are obtained for merging overlapping rules from base T rules, using all adaptive features. Best results for merging rules from NT rules are slightly lower.

Table 9 evaluates the same configurations of base rule cutoffs and features on the TED talk data. Here the best result is a nominal improvement of 0.3 BLEU points over the baseline, obtained by merging rules from base T rules. Again, this result is slightly better than merging rules from base NT rules. However, in case of the TED data, no result difference

	News Commentary	TED
MERT	24.95	25.94
PairRank	25.69†	25.90

Table 7: Baseline results for News Commentary and TED talks German-to-English translation. Statistically significant differences to MERT are denoted with †.

Cutoff	Features (1)-(2)	(1)-(3)	(1)-(8)	(9)-(16)
T 5	25.83	25.82	25.83	*25.86*
T 8	25.99	25.99	26.02	**26.24**†
T 11	25.93	26.08†	*26.12*†	25.75
NT 17	25.76	*26.13*†	26.01	25.84
NT 20	*26.14*†	25.70	25.89	25.97
NT 23	25.76	25.90	*26.22*†	25.82

Table 8: Results for News Commentary, German-English translation. Best results for a certain feature set in *italics*, best result overall in **bold**. Significant differences compared to the PairRank baseline of Table 7 are denoted with †.

Cutoff	Features (1)-(2)	(1)-(3)	(1)-(8)	(9)-(16)
T 5	25.89	25.94	25.93	*26.00*
T 8	25.96	25.95	**26.23**	26.01
T 11	25.84	*26.13*	25.79	25.98
NT 17	25.71	25.93	*26.04*	25.82
NT 20	25.50	26.03	26.02	*26.10*
NT 23	25.57	*26.04*	26.01	25.78

Table 9: Results for TED, German-English translation. Best results for a certain feature set in *italics*, best result overall in **bold**. Significant differences compared to the PairRank baseline of Table 7 are denoted with †.

is statistically significant compared to the PairRank baseline.

Table 10 shows an evaluation for a combination of overlapping phrase rules merged from base T rules and base NT rules. Combining the best configurations for generating overlapping phrases from T-only and NT base rules yields results that are about 0.1 BLEU point lower than the best results in Tables 8 and 9. Result differences are statistically significant for News Commentary, but not for TED experiments.

Overall, we find a correspondence of BLEU improvements shown in Tables 8, 9, 10 with the pattern of composition and usage shown in Table 5, with higher gains and higher usage for T rules compared to NT rules.

6. Conclusion

We presented an application of the idea of offline merging of bi-phrases into longer phrases with overlapping words to the framework of hierarchical phrase-based translation. The advantages of overlapping phrases in translation are enhanced fluency in positions that would otherwise be phrase boundaries. Furthermore, a large number of new phrases can be generated that have never been seen in the training data but are applicable to the test data. Our approach maintains all the benefits of using overlapping phrases at translation time, without the pain of having to modify the decoder to deal with overlapping phrases.

Our experimental results on two datasets for German-to-English translation show gains of $0.3-0.6$ BLEU points over a baseline system that implements discriminatively trained hierarchical phrase-based SMT. We conjecture that improved quality at translation time might be worth the overhead of building overlapping rules at phrase extraction time.

	News Commentary	TED
T + NT	**26.15**†	**26.10**

Table 10: Best results for combination of NT and T overlapping phrases on TED and News Commentary, German-English translation. Significant differences compared to the PairRank baseline of Table 7 are denoted with †.

7. References

[1] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, Edmonton, Cananda, 2003.

[2] M. Simard, N. Cancedda, B. Cavestro, M. Dymetman, E. Gaussier, C. Goutte, and K. Yamada, "Translating with non-contiguous phrases," in *Proceedings of the Human Language Technology Conference / Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, Vancouver, BC, Canada, 2005.

[3] M. Galley and C. D. Manning, "Accurate non-hierarchical phrase-based translation," in *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL (HLT-NAACL)*, Los Angeles, CA, 2010.

[4] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, 2007.

[5] M. Kääriäinen, "Sinuhe - statistical machine translation using a globally trained conditional exponential family translation model," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Singapore, 2009.

[6] B. Roth, A. McCallum, M. Dymetman, and N. Cancedda, "Machine translation using overlapping alignments and samplerank," in *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas (AMTA)*, Denver, CO, 2010.

[7] Z. Wang and J. Shawe-Taylor, "A kernel regression framework for SMT," *Machine Translation*, vol. 24, pp. 87–102, 2010.

[8] A. Tribble, S. Vogel, and A. Waibel, "Overlapping phrase-level translation rules in an SMT engine,"

in *Proceedings of the International Conference on NLP and Knowledge Engineering (NLP-KE)*, Beijing, China, 2003.

[9] C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, P. Blunsom, H. Setiawan, V. Eidelman, and P. Resnik, "cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models," in *Proceedings of the ACL 2010 System Demonstrations*, Uppsala, Sweden, 2010.

[10] C. Cortes, M. Mohri, and J. Weston, "A general regression framework for learning string-to-string mappings," in *Predicting Structured Data*, G. Bakhir, T. Hofmann, and B. Schölkopf, Eds. Cambridge, MA: The MIT Press, 2007, pp. 143–168.

[11] N. Serrano, J. Andrés-Ferrer, and F. Casacuberta, "On a kernel regression approach to machine translation," in *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*, Póvoa de Varzim, Portugal, 2009.

[12] E. Biçici and D. Yuret, "Regmt system for machine translation, system combination, and evaluation," in *Proceedings of the 6th Workshop on Statistical Machine Translation (WMT)*, Edinburgh, Scotland, UK, 2011.

[13] S. Vogel, Y. Zhang, F. Huang, A. Tribble, A. Venugopal, B. Zhao, and A. Waibel, "The CMU statistical machine translation system," in *Proceedings of MT Summit IX*, New Orleans, LA, 2003.

[14] R. D. Brown, R. Hutchinson, P. N. Bennett, J. G. Carbonell, and P. Jansen, "Reducing boundary friction using translation-fragment overlap," in *Proceedings of MT Summit IX*, New Orleans, LA, 2003.

[15] R. Hutchinson, P. N. Bennett, J. G. Carbonell, P. Jansen, and R. D. Brown, "Maximal lattice overlap in example-based machine translation," Computer Science Department, Carnegie Mellon University, Paper 324, Tech. Rep., 2003.

[16] Y. S. Chang, H. T. Ng, and D. Chiang, "Word sense disambiguation improves statistial machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, Prague, Czech Republic, 2007.

[17] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Baltimore, MD, 2014.

[18] A. Lopez, "Hierarchical phrase-based translation with suffix arrays," in *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, Prague, Czech Republic, 2007.

[19] M. Denkowski, C. Dyer, and A. Lavie, "Learning from post-editing: Online model adaptation for statistical machine translation," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL'14)*, Gothenburg, Sweden, 2014.

[20] P. Simianer, S. Riezler, and C. Dyer, "Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, Jeju, Korea, 2012.

[21] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of ibm model 2," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, Georgia, 2013.

[22] K. Heafield, "KenLM: faster and smaller language model queries," in *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation (WMT'11)*, Edinburgh, UK, 2011.

[23] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable modified kneser-ney language model estimation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL'13)*, Sofia, Bulgaria, 2013.

[24] P. Koehn, H. Hoang, A. Birch, C. Callison-Birch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the ACL 2007 Demo and Poster Sessions*, Prague, Czech Republic, 2007.

[25] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proceedings of the 10th conference on European chapter of the Association for Computational Linguistics (EACL'03)*, Budapest, Hungary, 2003.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," Yorktown Heights, N.Y., Tech. Rep. IBM Research Division Technical Report, RC22176 (W0190-022), 2001.

[27] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, Edmonton, Cananda, 2003.

[28] J. Clark, C. Dyer, A. Lavie, and N. Smith, "Better hypothesis testing for statistical machine translation: Controlling for optimizer instability," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL'11)*, Portland, OR, 2011.

[29] E. W. Noreen, *Computer Intensive Methods for Testing Hypotheses. An Introduction*. New York: Wiley, 1989.

[30] S. Riezler and J. Maxwell, "On some pitfalls in automatic evaluation and significance testing for MT," in *Proceedings of the ACL-05 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, Ann Arbor, MI, 2005.

TRANSLATIONS OF THE CALLHOME EGYPTIAN ARABIC CORPUS FOR CONVERSATIONAL SPEECH TRANSLATION

Gaurav Kumar[1], Yuan Cao[1], Ryan Cotterell[1], Chris Callison-Burch[2], Daniel Povey[1], Sanjeev Khudanpur[1]

[1]Center for Language and Speech Processing & HLTCOE, Johns Hopkins University, Baltimore, USA
[2]Computer and Information Science Department, University of Pennsylvania, Philadelphia, USA

ABSTRACT

Translation of the output of automatic speech recognition (ASR) systems, also known as speech translation, has received a lot of research interest recently. This is especially true for programs such as DARPA BOLT which focus on improving spontaneous human-human conversation across languages. However, this research is hindered by the dearth of datasets developed for this explicit purpose. For Egyptian Arabic-English, in particular, no parallel speech-transcription-translation dataset exists in the same domain. In order to support research in speech translation, we introduce the Callhome Egyptian Arabic-English Speech Translation Corpus. This supplements the existing LDC corpus with four reference translations for each utterance in the transcripts. The result is a three-way parallel dataset of Egyptian Arabic Speech, transcriptions and English translations.

Index Terms— Spoken Language Translation, Speech Recognition, Machine Translation, Language Resources, Corpus Creation

1. INTRODUCTION

Translation of the output of automatic speech recognition (ASR) systems, also known as speech translation, has been the subject of research for several years now. Major programs that focused on this were VERBMOBIL, NESPOLE!, DARPA TRANSTAC, DARPA GALE and the Quaero project. The early projects were limited domain and limited vocabulary systems built to cater to machine directed or well enunciated speech. However, DARPA GALE and Quaero required large vocabulary continuous speech recognition systems with generic language models for ASR, and wide coverage SMT systems for translation. As both ASR and statistical machine translation systems have become more effective over the years, speech translation has once again become a major topic of research. The focus of the most recent project, DARPA BOLT (similar to its predecessor DARPA GALE), is to build spoken language translation (SLT) systems for spontaneous, conversational, human-human speech. In contrast to machine directed or scripted conversations (broadcast news), most conversational speech has by nature, variability in recording environment and vocal registers and a high number of disfluencies and out-of-vocabulary words. It also exhibits difficult challenges associated with code switching and regional dialects. This directly relates to an increase of difficulty for both ASR and SMT systems. Since SLT systems are generally built by feeding the output of the ASR system to an SMT system, each trained on separate datasets [1, 2], errors produced by the systems compound.

With respect to Egyptian Arabic specifically, unscripted, spontaneous, telephone conversations have been available through the Callhome Egyptian Arabic corpus (speech and transcripts) since 1997. However, since this dataset did not come with translations for the transcriptions in Arabic, researchers had to resort to using out-of-domain data to train the SMT systems. Transcripts for spontaneous conversations (speech), vary significantly from transcripts for scripted conversations and informal written conversations (web, forum, SMS, chat).

To bridge this gap between the type of data the ASR and SMT systems were trained on for SLT applications, we have created the Callhome Egyptian Arabic Speech Translation dataset. This supplements the existing LDC corpus with four reference translations for each utterance in the transcripts. The result is a three-way parallel dataset of Egyptian Arabic Speech recordings, transcriptions of the Arabic speech, and translations into English.

The primary goal of this paper is to describe the process of creation of this corpus in time for its pending public release, so that researchers who use the corpus have a good understanding of both its scope and limitations. We believe that this corpus will enable considerable new research in translation of spontaneous/conversational Arabic speech into English.

This work was supported by NSF IIS award No 0963898 and DARPA BOLT contract No HR0011-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, DARPA or the U.S. Government.

LDC Catalog Number	Name	#train	#dev	#eval
LDC97S45, LDC97T19	Callhome Egyptian Arabic Speech/Transcripts	80	20	20
LDC2002S22, LDC2002T39	1997 HUB5 Arabic Evaluation	0	0	20
LDC2002S37, LDC2002T38	Callhome Egyptian Arabic Speech/Transcripts Supplement	0	0	20

Table 1. Sizes (in # conversations) of the Callhome Egyptian Arabic corpus, supplements and evaluation datasets. The conversations last between 5-30 minutes.

Partition	# Utt's	# Words	Words/Utt
ECA-96 (train)	20,861	139,035	6.66
ECA-96 (dev)	6,415	34,543	5.38
ECA-96 (test)	3,044	16,500	5.42
97-eval-H5	2,800	18,845	6.73
ECA-supplement	2772	18039	6.51

Table 2. Partition statistics for the Callhome Egyptian Arabic corpus, supplements and evaluation datasets. Column 2,3 and 4 represent number of utterances, numbers of words and average number of words per utterance respectively.

2. CORPUS AND TRANSLATION SETUP

We present English translations of the Egyptian-Arabic Callhome corpus, supplements and evaluation sets. These datasets were commissioned and used by the DARPA GALE (Global Autonomous Language Exploitation), DARPA EARS (Effective Affordable Reusable Speech-to-text) and the NIST HUB-5 LVSCR (Large Vocabulary Conversational Speech Recognition) programs.

The speech part of the corpus consists of unscripted telephone conversations between native speakers of Egyptian Colloquial Arabic (ECA). The conversations last between 5-30 minutes. In addition to the conversations, speaker metadata including gender, age, education and accent is available. Conversation metadata includes channel quality, crosstalk identifiers and number of speakers.

For each of the conversations, transcripts that cover a contiguous 5-10 minute segment are available. Manual audio segmentation information is available through the transcripts which have start and end time for each utterance in a conversation. Since ECA does not have a standard orthographic system, the conversations were transcribed using a romanized orthographic system which was phonemically based. This system preserves word pronunciation information and word identity. These transcripts in romanized orthography were then converted to Arabic script (encoding : ISO 8859-6) using a lexicon lookup [LDC99L22]. Table 2 provides details about this corpus.

2.1. Callhome Egyptian Arabic Speech/Transcripts

This corpus [Speech: LDC97S45, Transcripts: LDC97T19], hereafter referred to as ECA-96, consists of 120 unscripted telephone conversations. The corpus is split into three partitions : train, dev and eval, containing 80, 20 and 20 conversations respectively. The transcripts contain 30,320 utterances with a total of 190,078 words.

2.2. 1997 HUB5 Arabic Evaluation

This corpus [Speech: LDC2002S22, Transcripts: LDC2002T39], hereafter referred to as 97-eval-H5, was used as the evaluation set for the 1997 NIST HUB-5 non-English evaluation of conversational speech recognition systems. It consists of 20 unscripted telephone conversations. The transcripts contain 2,800 utterances with a total of 18,845 words.

2.3. Callhome Egyptian Arabic Speech/Transcripts Supplements

This corpus [Speech: LDC2002S37, Transcripts: LDC2002T38], hereafter referred to collectively as the ECA supplement, was initially sequestered for future NIST evaluations, but later released as a supplement to ECA-96. It consists of 20 unscripted telephone conversations. The transcripts contain 2,722 utterances comprised of 18,039 words.

2.4. Special Symbols in Transcription

Since the telephone conversations in this corpus are informal in nature and unscripted, special symbols are used to mark sections of the conversation that are not conventional Arabic speech. These contain non-verbal vocalizations, disfluencies, background noise and distortion. Table 3 provides a sample of some of these special symbols. Further details are available in the documentation of the respective corpus.

2.5. Egyptian Arabic Lexicon

An Egyptian Arabic colloquial pronunciation dictionary which supplements the corpora mentioned above, is available (LDC99L22). The lexicon contains 51,202 entries from the ECA-96, ECA-supplement and the Badawi and Hines dictionary of Egyptian Colloquial Arabic. This lexicon includes

Symbol	Interpretation
{text}	sound made by the talker
[text]	background or channel sound
<language text>	speech in another language
((text))	unintelligible, best guess provided
(( ))	unintelligible; can't guess text
text	idiosyncratic word, not in common use
-text , text-	partial words

Table 3. A sample of the special symbols using in the Arabic transcripts. These represent non-conventional speech segments such as non-verbal vocalizations, disfluencies, background noise and distortion.

orthographic representation of words in the LDC romanization scheme and Arabic script along with morphological, phonological, stress, source, and frequency information.

3. TRANSLATION METHODOLOGY

The translations for the Egyptian Arabic Callhome corpus were obtained using crowd-sourcing techniques. Crowdsourcing has become a standard technique in the collection and annotation of scientific data [3, 4, 5, 6, 7, 8] including data for natural language processing tasks like machine translation [9]. We use the crowdsourcing platform, Amazon Mechanical Turk (MTurk) to obtain translations. We follow the best practices suggested by [9] in this process.

3.1. Pre-processing

Each transcript was pre-processed to remove markup, including the special symbols described in Section 2.4. Some special symbols contain text in a foreign language (mostly, English). These were retained so that they could be passed through to the translation. Utterances that comprised only of markup and the special symbols were removed. Each utterance in the corpus contains channel and segment information. These were incorporated as a part of a segment identifier so that the translations could be mapped back to the transcriptions and the speech segments.

3.2. Collecting Translations

A translation task on the MTurk platform is presented to the translators as a HIT (Human Intelligence Task). Each translator was presented with a sequence of ten segments to translate. These segments or utterances were always presented in the order they appear on the transcripts. Since the conversation consists of two channels, the order presented generally comprised of alternating speakers. This allowed the translators to incorporate context wherever it is available and helpful. Each HIT included translation instructions derived from [9]. In addition, translators were instructed to retain the foreign language information in the utterance. As noted earlier, the transcripts were converted to Arabic script from an intermediate romanized version. We did not attempt to normalize any non-MSA words to create an MSA equivalent. Four independent translations were obtained for each utterance using such HITs.

3.3. Quality Control

MTurk provides a quality control mechanism which relies on vetting of users and qualification tests. However, these methods in isolation are not enough to guarantee high quality translations. We used the following quality control measures to ensure that the quality of the translations was acceptable and to prevent inappropriate use of the platform.

- For each utterance, we obtained translations from Google Translate. If a translation had a small edit distance from the translation obtained via Google Translate, it was flagged, reviewed and rejected if it had the same errors.
- The utterance for translation task was presented as an image rather than as text. This prevented users from using online translation services to cut and paste translations.
- Manually translated gold standard segments were inserted into our dataset. Each translator was presented with three such segments. Their HITs were flagged, reviewed and rejected if their translation for these segments was not similar to the gold standard translations.
- We gathered self-reported geographical and language information for each of our contributors on MTurk. The specification for our task asked native Arabic speakers to participate. Since HITs had to be manually approved, we checked translator metadata and number of translations received. In addition, prior to approval, a spot check of the translations was conducted. Finally, higher preference was given to trusted Arabic speakers that we have worked with on other translation tasks.

3.4. Post-processing

The translations were split based on the partitions described in Section 2 and each partition was duplicated (typically fourfold) to obtain redundant/independent translations. For some utterances, we ended up obtained more than four translations. These were stored in an overflow file. Utterances that only contained markup and special symbols (which were previously removed) were re-inserted into this set of translations to restore utterance-level synchronization with the LDC corpora.

Partition	# Utt's	# Words	Words/Utt
ECA-96 (train)	86,313	713,549	8.27
ECA-96 (dev)	25,769	186,400	7.23
ECA-96 (test)	12,212	85,182	6.98
97-eval-H5	11,248	91,647	8.15
ECA-supplement	11,126	87,489	7.86

Table 4. The results of the translation task described in section 4. Each utterance in the original partitions has about four redundant translations. The number of utterances in column 2 has hence effectively been multiplied by 4. The last column represents the number of words per utterance in the translations.

Partition	Crossfold BLEU
ECA-96 (train)	40.09%
ECA-96 (dev)	35.64%
ECA-96 (test)	35.86%
97-eval-H5	35.81%
ECA-supplement	37.15%

Table 6. Inter-annotator BLEU per partition of the Callhome Egyptian Arabic corpus, supplements and evaluation datasets. Each translation was evaluated against three translations to obtain a BLEU score per utterance. This was averaged per partition.

4. TRANSLATION TASK RESULTS AND CONSISTENCY

In total, 838 translators participated in this process, producing 143,568 translations in English. Table 4 summarizes the results of the translation task. Note that the average number of words per utterance has increased after translation to English. Table 5 provides a sample of the translations obtained.

To measure inter-annotator agreement, we used a cross-folding type BLEU scoring scheme. Translations for each partition were lower-cased, tokenized using the Penn WSJ treebank conventions, and punctuation was normalized. Each translation was then evaluated against the remaining three translations in a cross-folding fashion. The results were averaged per dataset partition. The results of these experiments are in Table 6.

5. PLANNED CORPUS RELEASE

In a manner similar to the previous work on speech translation of [10], based on the Spanish Fisher and Callhome corpora, we plan to provide Automatic Speech Recognition (ASR) output for the datasets in the Callhome Egyptian Arabic corpus. The ASR output will be provided in the form of OpenFST lattices, lattice oracles (paths that have the least word error rate in the lattice) and the 1-best output. This will effectively lead to the creation of a four-way parallel dataset with Egyptian Arabic speech, transcripts, ASR output and English translations. Our goal in providing the ASR output is to enable research in speech translation for Statistical Machine Translation (SMT) researchers as well.

6. CONCLUSION

We presented the Callhome Egyptian Arabic Speech Translation Corpus based on the Callhome Egyptian Arabic corpus, supplements and evaluation (HUB5) datasets. With the ASR output, the resulting speech translation corpus is a four-way parallel dataset with Egyptian Arabic speech, transcripts, ASR output (lattice, lattice oracle and 1-best) and translations. This in-domain dataset is an effort to aid research in translation of spontaneous, conversational speech with a long term goal of improving human-human conversation.

7. REFERENCES

[1] Oliver Bender Richard Zens, "The RWTH phrase-based statistical machine translation system," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2005, pp. 155–162.

[2] E. Matusov, S. Kanthak, and H. Ney, "Integrating speech recognition and machine translation: Where do we stand?," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*, May 2006, vol. 5, pp. V–V.

[3] Scott Novotney and Chris Callison-Burch, "Cheap, fast and good enough: Automatic speech recognition with non-expert transcription," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA, 2010, HLT '10, p. 207215, Association for Computational Linguistics.

[4] A Sorokin and D. Forsyth, "Utility data annotation with amazon mechanical turk," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08*, June 2008, pp. 1–8.

[5] Aniket Kittur, Ed H. Chi, and Bongwon Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2008, CHI '08, p. 453456, ACM.

[6] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng, "Cheap and fastbut is it good?: Evaluating non-expert annotations for natural language tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA,

Source	ما انتو مبتردوش على التليفون يبقى
Translation 1	you do n't reply to the phone
Translation 2	so you do n't answer the phone then
Translation 3	you do n't answer the phone it seems
Translation 4	because you do n't answer the call then
Source	مصعبان عليه نفسه كمان
Translation 1	he feels hard for himself too
Translation 2	he feel bad about himself
Translation 3	he feels sorry for himself too
Translation 4	i feel sorrow about his condition too

Table 5. A sample of the translations obtained using the translation task described in section 4. The translations are lower-cased, tokenized and punctuation has been normalized.

USA, 2008, EMNLP '08, p. 254263, Association for Computational Linguistics.

[7] Chris Callison-Burch, "Fast, cheap, and creative: Evaluating translation quality using amazon's mechanical turk," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, Stroudsburg, PA, USA, 2009, EMNLP '09, p. 286295, Association for Computational Linguistics.

[8] Gabriele Paolacci, Jesse Chandler, and Panagiotis G. Ipeirotis, "Running experiments on amazon mechanical turk," SSRN Scholarly Paper ID 1626226, Social Science Research Network, Rochester, NY, June 2010.

[9] Omar F. Zaidan and Chris Callison-Burch, "Crowdsourcing translation: Professional quality from non-professionals," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, Stroudsburg, PA, USA, 2011, HLT '11, p. 12201229, Association for Computational Linguistics.

[10] Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur, "Improved speech-to-text translation with the fisher and callhome translated corpus of spanish-english speech," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2013.

Improving In-Domain Data Selection For Small In-Domain Sets

Mohammed Mediani, Joshua Winebarger, Alexander Waibel

Karlsruhe Institute of Technology
Karlsruhe, Germany
firstname.lastname@kit.edu

Abstract

Finding sufficient in-domain text data for language modeling is a recurrent challenge. Some methods have already been proposed for selecting parts of out-of-domain text data most closely resembling the in-domain data using a small amount of the latter. Including this new "near-domain" data in training can potentially lead to better language model performance, while reducing training resources relative to incorporating all data.

One popular, state-of-the-art selection process based on cross-entropy scores makes use of in-domain and out-of-domain language models. In order to compensate for the limited availability of the in-domain data required for this method, we introduce enhancements to two of its steps.

Firstly, we improve the procedure for drawing the out-of-domain sample data used for selection. Secondly, we use word-associations in order to extend the underlying vocabulary of the sample language models used for scoring. These enhancements are applied to selecting text for language modeling of talks given in a technical subject area.

Besides comparing perplexity, we judge the resulting language models by their performance in automatic speech recognition and machine translation tasks. We evaluate our method in different contexts. We show that it yields consistent improvements, up to 2% absolute reduction in word error rate and 0.3 Bleu points. We achieve these improvements even given a much smaller in-domain set.

1. Introduction

The need for in-domain data in machine learning is a well-established problem and should be well motivated in previous papers (e.g [1]). We briefly observe, however, that across domains system performance is tied to the similarity between training and testing data. The testing data used for guiding system development is almost synonymous with in-domain data. It follows directly that training data should also resemble the in-domain as closely as possible. In-domain data however is also almost always the most limited kind. This necessitates supplementing it with out-of-domain or non-domain-specific data in order to achieve satisfactory model estimates.

In this paper we consider the training of language models for speech recognition and machine translation of university lectures, which are very domain-specific. Typically this means adapting existing systems to a new topic. Perhaps unique to this application is that the in-domain data for lectures is normally of a very small size. A one-hour lecture may produce under a thousand utterances and roughly ten thousand words. The necessity of rapid system development and testing in this context encourages us to limit training data size. What we want, then is a way to reduce large amounts of data and at the same time improve its relevance. Ideally we would also be able to do so using only a very small amount of in-domain data.

We improve the work of [2] by drawing a better representative sample of out-of-domain data and language model (LM) vocabulary. However, more centrally, we extend the work of [2] by using a word-association based on a broad definition of similarity to extend these language models. With this extension, we do not compare solely the exact matching words from in-domain and out-of-domain corpora, but also their semantically associated words. These semantic associations can be inferred, as in the example of this paper through the use of pre-existing non-domain-specific parallel and/or monolingual corpora, or through hand-made thesauri. Then with a small amount of in-domain data we use the aforementioned extended language models to rank and select out-of-domain sentences.

1.1. Previous Work

The starting point and reference of our work is that found in [2], which is to our knowledge one of the most recent and popular methods in a series of methods on data selection [3, 4, 5]. Their approach assumes the availability of enough in-domain data to train a reasonable in-domain LM, which is used to compute a cross-entropy score for the out-of-domain sentences. The sentence is also scored by another, out-of-domain LM resulting from a similar-sized random out-of-domain sample. If the difference between these two scores exceeds a certain threshold the sentence is retained, the threshold being tuned on a small heldout in-domain set. This approach can be qualified as one based on the perplexity of the out-of-domain data. The in-domain data used in [2] is the EPPS corpus, which contains more than one million sentences. This stands in contrast to the lecture case with very specific domains and very limited data sizes. The

authors report their results in terms of perplexity, for which their technique outperforms a baseline selection method by twenty absolute points. Their approach has been shown to be effective for selecting LM training data, at least from the perspective of a Statistical Machine Translation (SMT) system with a specific domain task [6, 7, 8]. We note that the main task of these systems was to translate TED talks.[1] The work in [2] was extended to parallel data selection by [9, 10]. However, the last work concludes that the approach is less effective in the parallel case.

The approach of differential LM scores used in the aforementioned papers has a long history in the information retrieval (IR) domain [11, 12]. However, only unigram language models are considered in the context of IR, since the order in this task is meaningless.

Enriching the LM capability by incorporating word relationships has also been proposed in IR and is referred to as a *translation model* therein [13, 14].[2] More closely related to our approach, [15] uses word similarities to extend LMs in all orders. They show that extended LMs with properly computed word similarities significantly improve their performance at least in a speech recognition task.

1.2. Area of Application

The translation of talks and lectures between natural languages has gained attention in recent years, with events such as the International Workshop on Spoken Language Translation (IWSLT) sponsoring evaluations of lecture translation systems for such material as TED talks. From the perspective of Automatic Speech Recognition (ASR), talks and lectures are an interesting domain where the current state of the art can be advanced, as the style of speaking is thought to lie somewhere between spontaneous and read speech.

As noted previously, university lectures in particular are very domain-specific and thus in-domain data tends to be quite limited. The typical approach for language modeling in such a scenario is to include as much data as possible, both in- and out-of-domain, and allow weighted interpolation to select the best mixture based on some heldout set. However, if a satisfactory method could be found to choose only those parts of the out-of-domain set most similar to the in-domain set, this would reduce the amount of necessary LM training data. Not only would this save training time, it would also produce LMs that are smaller and possibly more adapted to the task at hand.

We perform text selection using variations of our technique and train language models on the resulting selected data. These LMs are then evaluated in terms of their perplexity on a heldout set, the word-error-rate of a speech recogniser, and an SMT system using the LM. We also apply the technique of [2] to our selection task as a reference.

[1] `http://www.ted.com`

[2] Note that we will use the terms "translation model" and "lexicon" interchangeably throughout the paper.

1.3. Paper structure

The remainder of the paper is structured as follows. In section 2 we describe the theory behind our enhancements to the standard selection algorithm. First, we discuss our method of intelligently selecting the out-of-domain LM used for cross-entropy selection. Next, we discuss our experiments with a more careful selection of the cross-entropy in-domain and out-of-domain language model vocabularies. In section 3.1 we introduce our association-based approach. We describe how we compute lexicons and how we use them to extend the cross-entropy language models. The results of our experiments are presented in section 5. We end the paper with section 6 in which we draw conclusions and discuss future work.

2. Enhancements

2.1. Drawing an out-of-domain representative sample

In the cross-entropy method of [2] previously described in Section 1.1, the out-of-domain LM is taken simply as a random sample of the larger out-of-domain data upon which we do selection, *OD*. However, randomly-drawn text may represent both in-domain as well as out-of-domain data (*OD*). The out-of-domain LM should instead represent the kind of data which we seek to exclude from our selection. Since the in-domain data should be the furthest from the latter kind of data, we reasoned that the in-domain LM could be used to intelligently select the data for the out-of-domain LM. We do this by first scoring the sentences in *OD* with the in-domain LM for perplexity (with a closed vocabulary). As some of our data in *OD* comes from web crawls, the sentences with the highest perplexity are mainly "junk" coming from automatic text processors and/or converters. The sentences with the lowest perplexity are mostly in the in-domain set. Therefore we specify some range around the median perplexity (m) as being a legitimate region from which to select sentences for the out-of-domain LM. In our case we chose $m \pm 0.5m$ with m being the median perplexity. Then for our out-of-domain LM we randomly draw an appropriate number of sentences from this range. The probability of any particular sentence being drawn is proportional to its corresponding perplexity.[3]

2.2. Vocabulary selection

Intuitively, we could think of vocabulary words as indicators of the importance of a sentence. Words occurring with high frequency in both in- and out-of-domain data sets would be of lower interest. In contrast, words frequently encountered in the in-domain only indicate that the sentence is of high importance. It was not clear to us whether the words which are common in the out-of-domain only would be a negative indicator. That is why we experimented with different ways

[3] For the weighted random sampling without replacement, we use the algorithm described in [16]

for choosing the vocabulary on which the LMs are based. The first vocabulary is taken as the intersection of the in- and out-of-domain vocabularies $V_1 = voc\{ID\} \cap voc\{OD\}$. The second vocabulary incorporates the first and adds those words which occur with high frequency in the in-domain source only. This is $V_2 = V_1 \cup hf\{ID\}$. The third incorporates the second (and consequently the first,) adding those high-frequency words occurring only in the out-of-domain LM dataset. Thus $V_3 = V_1 \cup hf\{OD\}$ A visual representation of this scheme is depicted in figure 1.

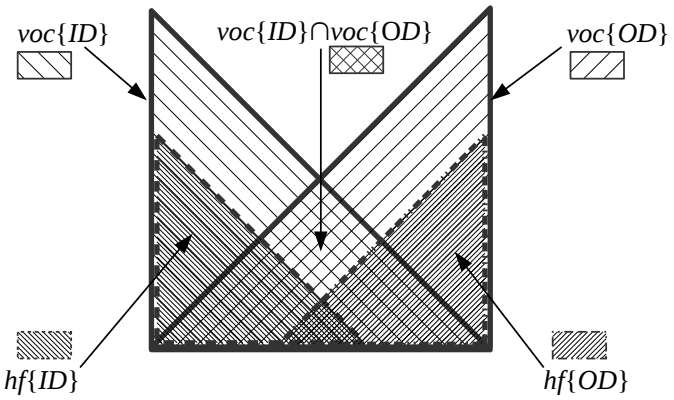


Figure 1: Diagrammatic representation of vocabularies of in- and out-of-domain sources

3. Extended Cross-Entropy Selection

In this section, we present our approach to create the word associations resulting in a lexicon quantifying the strength of relationships between vocabulary words and non-vocabulary words. First, the theoretical motivation for this kind of association is presented. Then the technical details on how our lexicon was built are discussed. Finally, the unigram LM extension is explained.

3.1. From bilingual word alignments to monolingual word associations

It is noteworthy that the lexical word-associations could be derived in many ways. These include manually hand-crafted thesauri (e.g. WordNet [17]) or automatically learned from monolingual corpora [18]. In this work, most of our experiments are based on lexicons derived form freely available parallel corpora, since we already dispose of relevant parallel data and computational tools to perform such a task.

Our lexicon derivation is based on the following assumption: In a perfectly aligned parallel corpus, words from the source language aligned to the same target word should be lexically related. Consequently, in creating a lexicon for a language (say, German) we infer associations between the (German) source words from their aligned target words (say, in English.) The association between two source words is proportional to the alignment probabilities relating them to the common target word.

Based on this assumption, we would like to estimate relationship strength (the so-called translation table) for pairs of words. One word of such a pair, the vocabulary word, is found in the LM vocabulary (and hence in the in-domain sample). The selection of this vocabulary is explained in Section 2.2. The other word comes from the source side (i.e. German) of the parallel corpus but is not present in the LM vocabulary.

Given a vocabulary word v and a non-vocabulary word w, the association $\mathrm{t}(w \mid v)$ is estimated as follows:

$$\begin{aligned} \mathrm{t}(w \mid v) &= \frac{\Pr(w, v)}{\Pr(v)} \\ &= \sum_z \frac{\Pr(z)\Pr(w, v \mid z)}{\Pr(v)} \\ &\approx \sum_z \frac{\Pr(z)\Pr(w \mid z)\Pr(v \mid z)}{\Pr(v)} \\ &= \sum_z \Pr(w \mid z)\Pr(z \mid v) \end{aligned} \tag{1}$$

In the second line of Equation (1), we rewrote the probability expression by introducing the aligned words z from the target side (i.e English) as a latent variable. In the third line, we simplified the expression in the previous line by assuming that source words are independent when conditioned on the target words.

3.2. Lexicon creation

We create our lexicon from automatically aligned parallel corpora (EPPS, NC, and Common Crawl). The corpora are preprocessed by removing obvious tokens which would not contribute to associating words such as numbers and punctuation marks. Then we use the Giza++ toolkit to train the IBM3 alignments in both directions (i.e German $\rightarrow$ English and English $\rightarrow$ German). We then symmetrize the resulting alignments using the intersection heuristic [19]. That is to say, we retain only alignment points which appear in both directions. An additional symmetrizing step we perform is removing links corresponding to a negative association.[4]

The resulting alignments allow us to compute the terms $\Pr(w \mid z)$ and $\Pr(z \mid v)$ in Equation (1) and therefore the lexicon.[5] The probabilities from this lexicon will be used to induce a likelihood for the words which do not occur in the original vocabulary of our LMs used for computing cross-entropy scores. We discuss this LM extension in Section 3.4.

[4]Two words x and y are negatively associated if $\Pr(x, y) < \Pr(x)\Pr(y)$ [20].

[5]In machine translation literature, the terms $\Pr(w \mid z)$ and $\Pr(z \mid v)$ are referred to as *Lexical Translation Models* (not to be confused with the model referred to as Translation Model in IR).

3.3. Associations from monoligual corpora

A more attractive approach to computing associations between words would be by exploiting monolingual resources. These are available in much more important quantities for any language compared to their parallel counterparts. We explored this approach by using the cosine similarity between word vectors returned by `word2vec` [21] to infer word associations. For each vocabulary word we include the 10 most similar non-vocabulary words in the resulting lexicon. The similarity score between a vocabulary word v and a non-vocabulary word w is computed as follows:

$$\mathrm{Sim}\,(w, v) = \frac{\mathbf{w} \cdot \mathbf{v}}{\|\mathbf{w}\|\|\mathbf{v}\|} + 1$$

where $\mathbf{w}$ and $\mathbf{v}$ are the word vectors associated with w and v respectively.

Then, the association $\mathrm{t}(w \mid v)$ is obtained by normalizing the similarity scores, as follows:

$$\mathrm{t}\,(w \mid v) = \frac{\mathrm{Sim}\,(w, v)}{\sum_{w'} \mathrm{Sim}\,(w', v)}$$

3.4. Extension of LMs

According to the cross-entropy selection, the out-of-vocabulary (OOV) words will have only a small effect on a sentence score. This is due to the fact that they are mapped to <unk> (the unknown word,) and therefore the probability returned from one model (e.g. the in-domain) cancels its counterpart from the other (e.g. the out-of-domain.)[6] Consequently, including more "important" words in the model with a realistic likelihood would conceivably make our model more robust.

To extend the LM with knowledge from the lexicon, we add to the unigram order those words which in the lexicon are associated with the LM vocabulary words. Therefore, these new unigrams can contribute to evaluating the sentence probabilities by the back-off mechanism. We found that the rate of backing-off to these new words is about 20%. The integration of the new unigrams is performed as follows. First, we discount the probabilities of the vocabulary words to free some a priori fixed mass (say $1 - m_0$.) Afterwards, each word added from the lexicon receives a share from m_0 proportional to two factors. The first factor is the LM probability of the associated vocabulary words. The second factor is the strength of the lexicon association connecting the out-of-vocabulary word to the in-vocabulary words. Note that m_0 is a tunable parameter. In our experiments, we found setting $m_0 = \Pr(<\text{unk}>)$ to be optimal.

Formally speaking, the probability of observing the word w given that the word sequence $w*$ is expressed as follows:

$$\Pr\,(w \mid w^*) = \begin{cases} m_0 \Pr_{LM}\,(w \mid w^*) & \text{if } |w^* w| > 0 \\ (1 - m_0) \sum_{v:|w^* v|>0} \mathrm{t}\,(w \mid v) \Pr_{LM}\,(v \mid w^*) & \text{otherwise} \end{cases}$$

[6] This effect will mostly be a penalization. In practice, the probability of <unk> is larger in the out-of-domain model

where w^* is an arbitrary sequence of words, possibly empty (for unigrams); $\Pr_{LM}$ is the original back-off LM probability; $|x^*|$ is the number of times the sequence x^* appears in the text; and t is the association table associating a vocabulary word v to a non-vocabulary word w. This procedure results in a new LM whose vocabulary is a superset of the original vocabulary. However, in most of this work we applied this extension at the unigram level only and hence kept the number of higher order n-grams unchanged.

4. Experimental Design

4.1. Data sources

For our out-of-domain data, we used a collection of monolingual German-language text corpora from various sources. This corpus totals around 37 million sentences and 0.67 billion tokens. We call this set of corpora *OD*. A table summarising these sources is given in table 1.

Type	Sentence count	Token count
News	11M	204M
Blog	3M	45M
Webcrawls	18M	345M
Parliamentary transcripts	256K	3.4M
Speeches and talks	6.8K	164K
Other sources	1.2K	18K
Total	37M	670M

Table 1: *Summary of monolingual out-of-domain text data used as a basis for data selection, which we term* OD

For bi-word association and lexicon training, we used a German-English parallel corpus we term *PC*. This consists of the public parallel corpora distributed for the WMT evaluation campaign [22] totaling 3.3 million lines of parallel text.

An in-domain corpus was available totaling 11 thousand lines and 237 thousand tokens, taken as mixture of transcriptions of several university lectures. We call this corpus *DEV*. Another similar-sized set from the same domain was held out in order to evaluate the perplexity of the resulting LMs.

For the purpose of computing ASR word error rates (WER), we took as a basis 16 hours of transcribed in-domain talk and lecture recordings from our in-house resources. The transcriptions for this set, composed of 13 thousand lines with 168 thousand tokens, were used as a set of held-out in-domain text for testing the perplexity of our language models. This held-out set is named *TEST*.

From the 16 hours of audio we randomly selected one hour on which to test the ASR. We call it *WERTEST*.

4.2. Selection Process

Our process of creating a set of selected texts from *OD* proceeded in several steps. Given *DEV* and *OD* we created an in-domain LM and out-of-domain LM. In our experiments with association-based scoring we extend the in-domain and out-of-domain LMs with information from our

lexicon. Next, scores were computed for each line in each source in *OD*. We then ranked all candidate lines across sources according to their score and retained only the top $K\%$ of candidates to carry over into the selected corpus *SEL*.

Our baseline experiments focused on creating selections from the base set, varying the top $K\%$ retained between 1% and 100%. After creating the set *SEL*, we performed some text normalisation such as compound word splitting.

German, the test language of our experiments, is known for use of compound words. As this makes contributes to a high out-of-vocabulary rate in ASR, a compound-splitting algorithm is typically employed in this field. For example "Entscheidungsfunktion" is split into "Entscheidungs+ Funktion." This algorithm requires a list of sub-words and selects the best split by maximizing the sum of the squared sub-word-lengths [23]. The *TEST* and *DEV* corpora are preprocessed using this technique, whereas as the alignment texts for the lexicon training are not. This necessitated the application of compound splitting after selection and prior to LM training.

5. Results

In this section, we compare the results of the different techniques mentioned in the previous sections (enhancements and extensions).

In our first sets of experiments as shown in Tables 2 and 3, we perform selection using a reasonably-sized in-domain set, *DEV*, with around eleven thousand sentences. [7] In Table 2 we report perplexity values of the LMs on *TEST*. For each selection technique we show the results of retaining either the top 1, 2, 5, 6, or 10% of sentences. The first row in the table is our baseline consisting of the state-of-the-art cross-entropy method of [2]. The improvements gained from the enhancements are shown in the second row. The remaining rows are related to applying the extension in different ways.

As shown in the third row, we apply the extension to the in-domain LM in the process of drawing the out-of-domain sample as explained in Section 2.1. For this we used only the high-frequency in-domain vocabulary, $hf\{ID\}$ as shown in Figure 1. After that, we retrain both the in- and out-of-domain LMs without extension. This configuration is referred to as "Extended Enhancements" (seen in the table as "Ext. Enhancements.")

In the fourth row we show the results of our "Extension" configuration. This configuration applies the extension only to our final in- and out-of-domain selection LMs (i.e., no extension was applied while drawing the out-of-domain sample), using the approach described in Section 3.4.

Finally, the previous two extensions are effectively combined. This means that we apply two independent extensions: we extend the in-domain LM in order to draw the out-of-domain representative and then we extend both in- and out-of-domain LMs for selection. We see this configuration, "Double Extension," on the fifth row of the table.

Table 3 shows WER resulting from using a subset of these LMs in a recognition task.

Technique	% Retained Sent. (ppl)				
	1	2	5	6	10
Moore, et al	222.7	202.4	190.3	190.0	190.5
Enhancements	211.9	195.4	185.3	184.5	185.9
Ext. Enhancements	208.1	192.9	183.4	183.3	185.0
Extension	206.2	191.9	183.0	182.5	184.4
Double Extension	203.0	189.1	181.3	181.0	183.3
Reference	% Retained Sent. (ppl)				
	100				
No selection	301.9				

Table 2: *Perplexity on TEST of the LMs selected using a reasonable in-domain set*

Technique	% Retained Sent. (WER)		
	1	5	10
Moore, et al	30.5	29.1	29.5
Enhancements	30.2	28.7	28.9
Double Extension	29.9	28.2	28.5
Reference	% Retained Sent. (WER)		
	100		
No selection	29.9		

Table 3: *Word error rate on WERTEST of LMs selected using a reasonable in-domain set*

In our second set of experiments, we simulated the case of hard conditions on the availability of in-domain data. We used a very small set of only one thousand sentences for our in-domain set as follows. First we split *DEV* into two parts, each part begin scored using the other. Then we merged them and selected the top-scoring one thousand sentences. This way, we assume that the resulting small set would be concentrated on the dominating topic of the whole set. The results of using this small in-domain set are summarized in Table 4.

We see that in the case of the small in-domain set, our method outperforms the baseline of [2] by between 40 and 60 perplexity points, and up to 2 percentage points absolute in terms of WER. For the reasonably-sized in-domain set, using enhancement alone gives larger gains than the incremental gains made by applying extension as well. For the small in-domain set, applying extension adds incremental gains comparable to the initial gains from enhancement.

Furthermore, we tested some of our selected data in a machine translation task. This is a phrase-based statistical system, where the translation model is trained on EPPS, NC, TED and BTEC English-German parallel corpora. It was tuned and tested on portions of a computer science lecture. The development set is around one thousand pairs whereas the test set is about two thousand. The weights of the log-linear model were tuned for a system using an LM trained on

[7] It is noteworthy that even this reasonably-sized in-domain set is less than 1% of the size of the in-domain set used in [2].

Technique	% Retained Sent. (ppl)		% Retained Sent. (WER)	
	5	10	5	10
Moore, et al	297.3	256.3	32.4	31.3
Ext. Enhancements	267.0	237.7	31.7	30.8
Double Extension	230.1	216.4	30.2	29.8

Table 4: *Perplexity and WER on TEST and WERTEST of LMs selected using a reduced in-domain set*

a completely different set. These were then kept unchanged for all tested models. The results of the translation experiments are shown in Table 5. Both enhancement and extension always outperformed the baseline. However for the cases of 10 and 20 percent retained sentences, the extension did not bring any additional gain.

Technique	% Retained Sent. (BLEU)		
	5	10	20
Moore, et al	13.24	13.04	12.84
Enhancements	13.47	13.19	13.06
Extension	13.52	13.16	13.00
Reference	% Retained Sent. (BLEU)		
	100		
No selection	12.47		

Table 5: BLEU scores for translation results

Finally, we performed some additional experiments in order to examine the extension in all ngram orders and the usage of associations induced from monolingual corpora. Table 6 shows the corresponding results. The first row repeats the last one in Table 2. The second row shows the results of a full extension, where we use the same principle as detailed in Section 3.4 in order extend words of the LM. However, here we extend all orders from 1 through 4 unlike the previous experiments where we only extended the unigrams. The results of monolingual-based associations are shown in the third row. In this case, the association is equivalent to the cosine similarity between word vectors (as explained in Section 3.3.) These vectors are computed using a large corpus (29 million sentences and 0.4 billion tokens). To do so, we use `word2vec` with *continuous bag of words* as the learning algorithm [21].[8] The size of the vectors is set to 500 and the context window to 10. Words appearing less than 5 times are discarded and the number of iterations used is 15.

It follows from these last experiments that both full extension and `word2vec` associations have no important effect on the performance. However, these can be considered as baselines for future experiments as they lack thorough hyper parameter tuning.

[8] `http://code.google.com/p/word2vec/`

Technique	% Retained Sent. (ppl)		
	1	5	10
Double Extension (only unigrams)	203.0	181.3	183.3
Full extension	203.0	181.4	183.4
`word2vec` associations	203.3	181.7	183.6
Reference	% Retained Sent. (ppl)		
	100		
No selection	301.9		

Table 6: *Perplexity on TEST of additional experiments*

6. Conclusion

We presented several extensions and enhancements to the state-of-the-art in-domain data selection method of [2]. Our techniques bring consistent improvements to the performance of the LM, given enough similarity between the test set and the set used for selection. Improvement is noticeable for a reasonably-sized in-domain set and it is quite more noticeable still for very small in-domain sets, where in terms of perplexity we substantially outperform the state-of-the-art. In both ASR and SMT scenarios, our techniques proved efficient by aggressively reducing the size of the training data. At the same time, they consistently improved the system's performance or in the worst case kept it unchanged.

While the automatically computed associations are cheaper to obtain, their hand-made counterparts are likely to be more accurate. Consequently, we plan to perform a comparison between these two for English, as it disposes of the largest hand-made thesaurus (WordNet).

It might be questioned why the associations used throughout this paper were inferred from general domain corpora, as this may lead to undesirable associations for a specific domain. Therefore, we would like to explore the effect of a pre-selection process over the data used to compute the association lexicon.

For the very small in-domain data sets, we think that better results could be obtained if one follows a bootstrapping strategy. That is, we repeatedly perform selection and add the best scoring sentences to the in-domain set and use the resulting set as the in-domain set for the next run.

We found both full extension and `word2vec` associations to be more expensive than the alignment-based unigram extension. Full extension suffers from a combinatorial explosion when the vocabulary size is reasonable. `word2vec` associations, on the other hand, are very slow to compute since we need to test each pair of words. We think we could improve this by performing the extension on a carefully selected subset from the vocabulary.

Another question we need to look into is the way we convert cosine similarities of `word2vec` into appropriate associations. The values we get from our current implementation are almost uniform. This might explain why this approach could not outperform the alignment-based associations, in spite of a much larger training corpus.

Lastly we close by noting that the tools developed for

lexicon creation are freely available on Github.[9]

7. Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658. The authors would also like to thank Jan Nieheues, Yuqi Zhang, and Ahmed Abdelali for their review and constructive comments.

8. References

[1] I. H. Daumé and D. Marcu, "Domain Adaptation for Statistical Classifiers," *J. Artif. Int. Res.*, vol. 26, no. 1, pp. 101–126, May 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622559.1622562

[2] R. C. Moore and W. D. Lewis, "Intelligent Selection of Language Model Training Data," in *ACL (Short Papers)*, 2010, pp. 220–224.

[3] D. Klakow, "Selecting articles from the language model training corpus," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 3. IEEE, 2000, pp. 1695–1698.

[4] S.-C. Lin, C.-L. Tsai, L.-F. Chien, K.-J. Chen, and L.-S. Lee, "Chinese language model adaptation based on document classification and multiple domain-specific language models," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[5] J. Gao, J. Goodman, M. Li, and K.-F. Lee, "Toward a unified approach to statistical language modeling for Chinese," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 1, no. 1, pp. 3–33, 2002.

[6] N. Durrani, B. Haddow, K. Heafield, and P. Koehn, "Edinburghs machine translation systems for European language pairs," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, 2013, pp. 112–119.

[7] J. Wuebker, M. Huck, S. Mansour, M. Freitag, M. Feng, S. Peitz, C. Schmidt, and H. Ney, "The RWTH Aachen machine translation system for IWSLT 2011," in *Proceedings of IWSLT*, 2011, pp. 106–113.

[8] T.-L. Ha, T. Herrmann, J. Niehues, M. Mediani, E. Cho, Y. Zhang, I. Slawik, and A. Waibel, "The KIT translation systems for IWSLT 2013," in *Proceedings of IWSLT*, 2013.

[9] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 355–362.

[10] S. Mansour, J. Wuebker, and H. Ney, "Combining translation and language model scoring for domain-specific data filtering," in *Proceedings of IWSLT*, 2011, pp. 222–229.

[11] J. Lafferty and C. Zhai, "Document Language Models, Query Models, and Risk Minimization for Information Retrieval," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '01. New York, NY, USA: ACM, 2001, pp. 111–119. [Online]. Available: http://doi.acm.org/10.1145/383952.383970

[12] W. Kraaij and M. Spitters, "Language Models for Topic Tracking," in *Language Models for Information Retrieval*, B. Croft and J. Lafferty, Eds. Kluwer Academic Publishers, 2003. [Online]. Available: http://www.springeronline.com/sgw/cda/frontpage/0,11855,5-153-22-33670504-detailsPage%253Dppmmedia%257Ctoc%257Ctoc,00.html

[13] A. Berger and J. Lafferty, "Information Retrieval As Statistical Translation," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99. New York, NY, USA: ACM, 1999, pp. 222–229. [Online]. Available: http://doi.acm.org/10.1145/312624.312681

[14] G. Cao, J.-Y. Nie, and J. Bai, "Integrating Word Relationships into Language Models," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '05. New York, NY, USA: ACM, 2005, pp. 298–305. [Online]. Available: http://doi.acm.org/10.1145/1076034.1076086

[15] I. Dagan, L. Lee, and F. C. N. Pereira, "Similarity-based models of word cooccurrence probabilities," *Machine Learning*, vol. 34, no. 1-3, pp. 43–69, 1999. [Online]. Available: http://dx.doi.org/10.1023/A:1007537716579

[16] P. S. Efraimidis and P. G. Spirakis, "Weighted random sampling with a reservoir," *Information Processing Letters*, vol. 97, no. 5, pp. 181– 185, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002001900500298X

[17] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748

[9] https://github.com/medmediani/pdict

[18] P. D. Turney, P. Pantel, *et al.*, "From frequency to meaning: Vector space models of semantics," *Journal of artificial intelligence research*, vol. 37, no. 1, pp. 141–188, 2010.

[19] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 48–54. [Online]. Available: http://dx.doi.org/10.3115/1073445.1073462

[20] S. Evert, "The statistics of word cooccurrences," Ph.D. dissertation, University of Stuttgart, 2004.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[22] "ACL 2014 Ninth Workshop on Statistical Machine Translation, Results and Collected Judgments," http://www.statmt.org/wmt14/translation-task.html, accessed: 2014-07-20.

[23] T. Marek, "Analysis of german compounds using weighted finite state transducers," *Bachelor thesis, University of Tübingen*, 2006.

MULTILINGUAL DEEP BOTTLE NECK FEATURES A STUDY ON LANGUAGE SELECTION AND TRAINING TECHNIQUES

Markus Müller[⋆]*, Sebastian Stüker*[⋆]*, Zaid Sheikh*[†]*, Florian Metze*[†] *and Alex Waibel*[⋆†]

International Center for Advanced Communication Technologies
[⋆]Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Germany
[†]Language Technologies Institute, Carnegie Mellon University, Pittsburgh PA, U.S.A.

ABSTRACT

Previous work has shown that training the neural networks for bottle neck feature extraction in a multilingual way can lead to improvements in word error rate and average term weighted value in a telephone key word search task. In this work we conduct a systematic study on a) which multilingual training strategy to employ, b) the effect of language selection and amount of multilingual training data used and c) how to find a suitable combination for languages. We conducted our experiment on the key word search task and the languages of the IARPA BABEL program. In a first step, we assessed the performance of a single language out of all available languages in combination with the target language. Based on these results, we then combined a multitude of languages. We also examined the influence of the amount of training data per language, as well as different techniques for combining the languages during network training. Our experiments show that data from arbitrary additional languages does not necessarily increase the performance of a system. But when combining a suitable set of languages, a significant gain in performance can be achieved.

Index Terms— bottle neck features, multilingual acoustic modeling, low-resource ASR, time-delay neural networks, data selection

1. INTRODUCTION

The goal of IARPA's program *BABEL*[1] is to build systems for keyword search (KWS) in telephone speech in a rapid manner and on limited amounts of data. Within the program progress is measured through annual evaluations. For the primary condition of the evaluation at the end of second year performers were only allowed to use 10h of transcribed data in the target language.

Since state-of-the-art key word search systems make use of *Large Vocabulary Continuous Speech Recognition* (LVCSR) systems, the task of rapidly building KWS systems includes the task of rapidly building LVCSR systems.

Building LVCSR systems for a new language requires large amounts of data in the target language in order to estimate the system's model parameters in a robust way. While the Babel evaluation's primary condition only allows for using data from the target language, another condition exists in which participants are allowed to use any data available within the BABEL program from any language in addition to the limited data of the target language.

In previous work we have shown for the Babel task that using multilingual data for training the neural network of the bottle neck feature (BNF) component of the pre-processing of the LVCSR system can either reduce training time [1] or the system's word error rate (WER) [2]. To the best of our knowledge, there has not been a concise analysis about which languages and data to choose. We therefore conducted a detailed study of how to combine the different languages within the BABEL program to improve a system given a specific target language.

In this paper we conduct a systematic series of experiments on training multilingual BNFs for the Babel task studying three aspects: a) which is the best technique for training the multilingual BNFs, b) is it more important to increase the total amount of training data or to vary the number of languages in BNF training, c) which is the best selection of languages for multilingual training.

The rest of the paper is structured as follows. In Section 2 we review related work. Then in Section 3 we describe how we trained our DBNFs. Section 4 describes our experimental set-up while Section 5 presents our experimental results.

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. This effort uses the IARPA Babel Program language collection releases IARPA-babel{102b-v0.4,103b-v0.3,101b-v0.4c,201b-v0.2b,203b-v3.1a,104b-v0.4bY,106-v0.2f,204b-v1.1b,105b-v0.4,107b-v0.7,206b-v0.1d}.

[1]http://www.iarpa.gov/index.php/research-programs/babel

2. RELATED WORK

2.1. Bottle Neck Features Extracted via Deep Belief Neural Networks

State-of-the-art LVCSR systems often use *deep belief neural networks* (DNNs) [3] for extracting features with the help of *deep belief bottle neck features* (DBNFs) [4, 5, 6]. For DBNFs a deep-belief network with several hidden layers and one bottleneck layers is trained, that classifies extracted feature vectors as, e.g., phonemes, context-dependent phonemes, or even model states. The layers of the DBNF are usually pre-trained, either by using *Restricted Boltzmann Machines* (RBMs) [4] or denoising auto-encoders [7]. After that, back-propagation training, in one of several possible variants, e.g., stochastic gradient descent combined with mini-batch training, is applied which we will call *fine-tuning* in this paper.

Past research has also shown that the use of *Time Delay Neural Networks* (TDNNs) for DBNF front-ends, which we sometimes also call shifting DBNFs, leads to performance improvements over the standard feed-forward networks used for DNNs [2].

2.2. Multilingual DBNFs

Recently the concept of multilingual acoustic modeling has also been extended to feature extraction via DBNFs. This is motivated by the fact that neural networks have been shown to be good at learning shared hidden representations across different tasks. With respect to multilingual modeling for DBNFs, the different languages that might partly overlap and partly differ in their phoneme inventory correspond to the different tasks, while the aspects common to the sounds across languages are the hidden aspects learned by the network. E.g., [8] has shown that the pre-training stage of the training of DBNFs is language independent. Training multilingual DBNFs can either be done by using one shared phoneme set [9], as it is done for ML-Mix [10], or by using different language dependent output layers, one for every language [11, 12, 13, 14]. The latter is possible, as the output layer is later discarded anyway and only the bottle neck layer of the network is retained for constructing the final feature vector. And just like the regular DBNFs, DBNFs using TDNNs can also be trained in multilingual fashion and lead to performance improvements [2].

Several strategies combining data from different languages have been explored. Thomas et al. analyzed the influence of varying the amount of data used from the target language in [15]. They built a multilingual system using a fixed amount of data from two training languages and studied the influnce of adding different amounts of data from the target language. Knill et al. merged the data from different languages during training thus creating a common phoneme set in [16]. They used a fixed set of multilingual data for training the acoustic model as well as the neural network and obtained an increase in performance. Grezl et al. used a similar approach in [17]. They trained a neural network using a fixed set of multilingual training data in combination with a limited set of data from the target language for adaptation.

While the work cited here has shown that multilingual training of DBNFs can lead to performance improvements, to the best of our knowledge no systematic study has been conducted that answers the three questions we aim to answer in this paper: a) is it more important to have more training data or to vary the number of languages in training, b) what is the best combination of multilingual training and the two stages of pre-training and fine tuning in DBNF training, c) and how important is the selection of languages when performing the multilingual training.

3. DEEP BOTTLE NECK FEATURES

3.1. Input Features for the DBNF Neural Network

There are several approaches towards the preprocessing the audio data before feeding it into the DBNF network. Among them are features such as mel-scaled cepstral coefficients (MFCC) and logarithmic mel-scaled spectral coefficients (lMel). Preliminary experiments have shown, that the use of MFCCs and lMels lead to similar results. We thus decided to use only lMels for our experiments. In addition to lMel features, we also include features derived from the fundamental frequency variation [18] and a pitch tracker[19]. These features are then combined and used as input to the DBNF neural network.

3.2. Deep Bottle Neck Features

The use of Deep Bottle Neck features as part of a speech recognition system has first been described by Grézl at al. in [20]. The common approach is to use a feed-forward neural network which is trained as a discriminative feature extractor. Our network contains a narrow, so called *bottle neck*, hidden layer. That layer consists of only a fraction of neurons in comparison to the other layers.

This network performs a non-linear discriminative dimensionality reduction. It has been shown that the activation of the bottle neck units are well suited as input features for HMM/GMM systems leading to improved recognition accuracy. In our set-up, we pre-train the network in an unsupervised fashion using denoising auto-encoders, and fine-tune the network via mini-batch training using stochastic gradient descent, adapting the learning rate via the new-bob algorithm [7].

3.3. Time-delay Neural Network Features

There are various methods and training strategies for neural networks. In a study conducted earlier[2] it as been shown that the use of TDNNs, which we sometimes call Shifting

Deep Bottle Neck features (SDBNFs) leads to improvements in performance. SDBNFs are based on the idea of a time delay neural network[21]. The concept of this approach is to bring the stacking of input features to the neural network level. Here, in the forward pass of the fine tuning step, the gradients of adjacent frames are being averaged over a window of several frames, thereby capturing the information over a longer period in time than just a single frame [2].

3.4. Multilingual Deep Bottle Neck Features

Data from multiple languages can be included at various stages of the neural network training. The first step in which data can be added is the pre-training. [8] showed that the use of multilingual data in an unsupervised way can be beneficial. In our experiment, the data we use from different languages has similar properties. It was recorded under the same recordings conditions. Therefore, the network can learn to extract features from human speech recorded under similar conditions in a language independent way. The role of the pre-training is to initially guide the network parameters into the right direction prior to the fine-tuning. By using more data, the network has the ability to generalize more due to the fact that the network parameters can be estimated in a more robust way.

The fine-tuning takes place as a second step. It offers the possibility to add data from multiple languages as well. Like when pre-training a network, here we also have the opportunity to include data from other languages as well. The same holds true when applying the shifting step. But since these steps fine-tune the parameters of the network, they are somewhat more language-dependent as they need to extract features resembling the individual sounds of a language. Thus, the question remains at which stages to work with multilingual data and at which only with the target data.

4. EXPERIMENTAL SETUP

We conducted our experiments with the help of the Janus Recognition Toolkit (JRTk) [22] which features the IBIS decoder [23]. As target language in our experiments we used Tamil, for which we trained speech recognition systems using different kinds of multilingual DBFNs. In our experiments, the DBNFs are the only part trained multilingually. The HMM/GMM system itself is only trained on the LLP dataset (10h) from Tamil. As we wanted to focus our study to the DBNF component of the system, we kept everything else fixed.

We assessed the performance of the systems on the development data set provided for Tamil. It consists of 10 hours of audio data. The systems were evaluated using two different metrics: Word error rate (WER) and average term weighted value[24] (ATWV). The latter requires a set of keywords; for this we used the given development keyword list. ATWV gives scores in the range between 0 and 1. For better readability, we multiplied the ATWV score by 100. We used a class based language-model and an automatic segmentation. Throughout our experiments, we keep the decoding parameters identical.

4.1. Corpora

We used data from the IARPA BABEL project. The IARPA provided data for several languages. These are: Assamese, Bengali, Cantonese, Haitian Creole, Lao, Pashto, Tagalog, Tamil, Turkish, Vietnamese and Zulu. Table 1 provides an overview of all the languages used, including details about the language family and the phoneme inventory. The languages selected for the BABEL program cover a wide variety of different language families. The number of phonemes per language ranges from 32 (Haitian Creole) to 68 (Vietnamese). As Tamil is the target language in our experiments, we also looked into the amount of phonemes that Tamil shares with each language. This information is presented in the last column of table 1

Language	**Language Family**	**# Ph.**	**# Ph. w. T.**
Tamil	Dravidian	34	-
Assamese	Indo-European	50	20
Bengali	Indo-European	51	21
Hait. Creole	(French) Creole	32	17
Lao	Tai-Kadai	41	20
Pashto	Indo-European	43	24
Tagalog	Austronesian	46	20
Turkish	Turkic	41	25
Vietnamese	Austroasiatic	68	18
Cantonese	Sino-Tibetan	37	14
Zulu	Niger-Congo	47	16

Table 1: Language overview, including the **language family**, size of **ph**oneme set and amount of **ph**onemes that each language shares **w**ith **T**amil

For each language, two data sets were provided: a limited language pack (LLP) and a full language pack (FLP). The LLP of a language consists of 10h of transcribed conversational speech. The FLP of a language consists of approximately 100h of transcribed data and includes the data from the LLP. The data itself is mainly narrowband telephone speech sampled with 8kHz. Some languages from the second year of the project (Assamese, Bengali, Haitian Creole, Tamil and Zulu) include some recordings with higher, CD-quality resolution. For our experiments, we resampled those down to 8kHz. The recordings contain different types of noises, as they were performed on the street, while driving a car or in an office with some machinery running in the background.

4.2. Baseline

The target language for our experiments is Tamil. For our baseline, we trained a system on the LLP dataset from Tamil only. First, we built a context-independent system from scratch using a flatstart approach. On top of that, we built a context-dependent system with 2,000 models.

Using that context-dependent system, we created the data required to train a DBNF. Our DBNFs consists of five hidden layers. With the exception of the bottle neck layer, each layer consists of 1,000 neurons. The bottle neck layer has only 42 neurons. For pre-training, we are using denoising auto-encoders with Gaussian noise and a corruption rate of 20%. To extract training data from the other languages for the neural network training, we used the FLP per language to train systems in a similar manner and to create that data.

In order to create forced alignments for languages other than Tamil, we trained a context-dependent system on the FLP dataset of that particular language. For selecting different amounts of training data, we randomly choose sets of speakers resembling the defined amount of audio data.

5. RESULTS

In our results we examined three different aspects: a) whether it is more important to use more data in multilingual DBNF training, or whether it is more important to have data from more, different languages; b) at which stages in the training of DBNFs is multilingual training data helpful; c) how to select the languages from which to train the DBNF for a specific language.

Therefore, we initially conducted an analysis to determine the performance of data from a single language in combination with the target language in Section 5.1. Parallel to that, we varied in Section 5.2 the amount of data for a selection of languages. We also investigated the use of additional language data at different points of neural network training in Section 5.3. Finally, we combined the best fitting languages together and as a contrastive experiment the worst fitting languages in Section 5.4.

5.1. Combination of a Single Language with Tamil

In order to establish a baseline for multilingual DBNFs we trained multilingual DBNFs on only two languages, by combining the data from the Tamil LLP with 40h of one other language. This will give a first impression of the usefulness of adding training data from other languages and will show the variance in performance depending on the exact language that was chosen to be added. We compare the resulting WERs against a baseline in which the DBNF was trained on Tamil LLP only. For this experiment, we used the multilingual data during pre-training, fine-tuning and the shifting step.

The results are shown in Table 2. One can see that the choice of language is important for the performance of the resulting DBNF. Some combinations lead to better performance, while others decrease the performance. The best results can be archived using Turkish, Pashto or Haitian Creole where we see gains of up to 1.6% relative in terms of WER over the monolingual baseline. Similar gains can be observed for ATWV. Here the best system (Turkish) improves from 2.67 to 3.96. However in the worst case, when choosing Vietnamese as additional language, the WER increases by 4.7% relative, while ATWV drops to -1.34.

The gains and losses correspond to some degree with the amount of shared phonemes between Tamil and each language. As shown in Table 2 the best fitting languages (Turkish and Pashto) share the largest amount of phonemes with Tamil, whereas the worst fitting languages (Vietnamese, Cantonese and Zulu) share the least phonemes with Tamil. But the amount of shared phonemes should only be considered as an approximation of the expected performance gain since for example Haitian Creole fits equally well to Tamil like Bengali and Pashto, although it only shares 17 phonemes with the target language.

Language	WER	ATWV	# Ph. w. T.
Baseline	82.6	2.67	-
+ Assamese	82.7	3.00	20
+ Bengali	81.5	3.26	21
+ Hait. Creole	81.5	3.82	17
+ Lao	82.3	2.97	20
+ Pashto	81.5	3.48	24
+ Tagalog	82.0	3.40	20
+ Turkish	**81.3**	**3.96**	**25**
+ Vietnamese	*86.5*	*-1.34*	*18*
+ Cantonese	83.3	1.53	14
+ Zulu	84.6	-0.04	16

Table 2: Tamil LLP plus additional 40h of another language. The last column shows the amount of **ph**onemes that each language shares **w**ith **T**amil

5.2. Varying the Amount of Additional Data

In the next experiment, we looked into varying the amount of foreign language data to Tamil LLP. Just as in the experiment before we added only one language to the Tamil training data, but this time either added the FLP (ca. 100h), 40h or 10h of training data of that language. We performed these experiments with the languages from the second year of the BABEL program. We added the language data to the whole training process of the neural network, including the pre-training, fine-tuning and shifting step.

Language	FLP	40h	10h
Assamese	82.4 / 2.54	82.7 / 2.37	82.0 / 3.28
Bengali	82.0 / 2.61	81.5 / 3.26	81.7 / 3.03
Hait. Creole	82.2 / 2.30	81.5 / 3.82	81.6 / 3.14
Lao	82.5 / 2.20	82.3 / 2.97	81.6 / 3.31

Table 3: Use of different amounts of data in combination with Tamil LLP. The number on the left denotes WER, the one on the right ATWV.

Table 3 shows the performance of the resulting systems. The results show that selecting the right amount of training data in addition to the 10h of Tamil training data is also important. Using all available data per language leads to performance degradation over the baseline. Matching the 10h of Tamil data with 10h of data from another language always leads to improvements over the baseline. For two out of the four languages taking 40h instead of 10h improves system performance even further, while for the other two languages this seems to be already too much training data, as performance starts to degrade again.

5.3. Methods of Using Data from Additional Languages

There are several steps in the training process of the neural network at which training data is used and therefore data from multiple languages can be added. Our training setup for neural networks consists of up to four steps: Pre-training, fine-tuning, shifting and again a fine-tuning step. For this experiment, we trained the networks used in three different ways, whereas multilingual data is being used in more and more steps: a) Using multilingual data only during pre-training, then performing the fine-tuning and shifting using data from Tamil LLP only. b) Using multilingual data for pre-training, fine-tuning and shifting. c) Using multilingual data for pre-training, fine-tuning, shifting and adding a fine-tuning step using data from Tamil LLP only.

This time we also looked at not only adding one language, but multiple languages to the DBNF training. We used 40h of data per language and the LLP for Tamil. Again, as in the previous experiment, we used data from the Babel second year languages Haitian Creole, Lao, Assamese and Bengali. This results in up to 160h of training data in addition to the 10h of data from the target language Tamil.

	H	H+L	H+L+A	H+L+A+B
a)	82.5 / 2.18	83.3 / 1.47	82.3 / 2.93	82.2 / 2.42
b)	81.5 / 3.82	81.2 / 3.63	80.8 / 4.06	80.6 / 4.05
c)	81.2 / 3.85	80.7 / 4.13	80.8 / 4.34	79.9 / 5.05

Table 4: Tamil LLP plus additional languages (**H**aitian Creole, **L**ao, **A**ssamese and **B**engali) and training methods: a) ML pre-training, b) ML pre-training and shifting, c) additional fine-tuning on Tamil LLP after shifting. The number on the left denotes WER, the one on the right ATWV.

As shown in Table 4, using the multilingual data in set-up a) (only for pre-training) yields only small gains, if at all. Set-up b) (using the multilingual data not only during pre-training, but also for fine-tuning and shifting) results in a gains of performance in all cases. The WER decreases up to 2.5% relative and the ATWV increases by 2.16. After applying another round of fine-tuning on Tamil only, and thus resulting in set-up c), performance is increased even further, by 0.5% relative in Terms of WER, and 1.0 in terms of ATWV.

As a general result, using additional data at all steps of the neural network training increases the performance the best. Likewise does an extra fine-tuning step on the target language after the multilingual training improve system performance further.

5.4. Combining Multiple Languages

Following our initial experiments using multiple languages and determining the performance of individual languages in combination with Tamil, we created two sets of four languages to do a first investigation into the best combination of multiple languages. One set consists of the best four languages according to Section 5.1. The second set contains the worst four languages. The languages are listed in Table 5.

Best fitting	Worst fitting
Turkish	Vietnamese
Hait. Creole	Zulu
Pashto	Cantonese
Bengali	Assamese

Table 5: Overview of languages fitting best and worst to Tamil. The best fitting languages are sorted starting with the best fitting language, the worst fitting languages are starting with the worst fitting language.

For this experiment, we use two schemes to determine the performance of the combination of the different languages. First, we used 40h per language, resulting in an additional amount of data of 40h, 80h, 120h and 160h of speech data. In a second set of experiments, we kept the amount additional data fixed to 40h, resulting in an amount of data per speech

depending on the number of additional languages. This results in an amount of 40h, 20h, 13h and 10h of additional data per language .

Language	40h p. l.	40h total	# Ph. w. T.
Baseline	82.6 / 2.67	82.6 / 2.67	-
T	81.3 / 3.96	81.3 / 3.96	25
T+H	81.0 / 4.50	80.9 / 4.21	25
T+H+P	80.3 / 5.41	80.5 / 4.66	28
T+H+P+B	79.7 / 5.65	79.9 / 5.52	28

Table 6: Use of additional languages (**T**urkish, **H**aitian Creole, **P**ashto and **B**engali) with either 40h of data **p**er **l**anguage or **40h** in **total** for all additional languages. The number on the left denotes WER, the one on the right ATWV. The last column shows the amount of **ph**onemes shared **w**ith **T**amil.

The results for combining the best systems are shown in Table 6. Integrating the best four languages into system training in addition to Tamil LLP decreases the WER by 3.6% relative and improves ATWV by 2.98. The results show that the more languages one adds, the better the performance gets. The difference between using 40h per additional languages and 40h of additional data in total is marginal. We see this as an indicator that the total amount of data used is not as important as the variety in the languages used for the multilingual training. Thus, when faced with the question of whether it is better to collect more data in few languages or more languages with fewer data, it is better to go for language diversity. The amount of shared phonemes corresponds here to some extent to the gain in recognition performance. Combining multiple languages increases the phoneme coverage of the target language.

Language	40h p. l.	40h total	# Ph. w. T.
Baseline	82.6 / 2.67	82.6 / 2.67	-
V	86.5 / -1.34	86.5 / -1.34	18
V+Z	82.4 / 1.98	82.5 / 1.91	20
V+Z+C	82.0 / 3.05	81.9 / 3.19	22
V+Z+C+A	81.6 / 3.90	81.7 / 3.85	24

Table 7: Use of additional languages (**V**ietnamese, **Z**ulu, **C**antonese and **A**ssamese) with either 40h of data **p**er **l**anguage or 40h in total for all additional languages. The number on the left denotes WER, the one on the right ATWV. The last column shows the amount of **ph**onemes shared **w**ith **T**amil.

When comparing the results of adding the best languages in Table 6 against the results in Table 7 where we added the worst languages, one sees that in the end, when adding enough languages, also adding bad performing languages gives gains over the baseline. Adding more languages increases the coverage of the phonemes here as well. The observed gain is comparable to using Pashto alone which has the same amount of shared phonemes with Tamil.

As described in section 2.2, other have also made use of the different phoneme sets by combining them. Although we compare the different phoneme sets, our system uses only the phonemes from the target language. We do not use the phonemes from the other languages explicitly. They were only used implicitly during the network training as target states while fine-tuning the network.

6. CONCLUSION AND OUTLOOK

In this work we have examined the use of multilingual DBNFs for Tamil speech recognition on the BABEL task. We have performed experiments to give insight into three questions: a) which is the best technique for training the multilingual DBNFs, b) is it more important to increase the total amount of training data or to vary the number of languages in DBNF training, c) which is the best selection of languages for multilingual training.

The experiments show that using multilingual data at all stages of our DBNFs (pre-training, fine-tuning, shifting stage) gives the best performance. Also, the total amount of training data is not as important as the variety of the languages in the multilingual training dataset. Some experiments suggest that adding too much data might from a certain point on decrease system performance again.

With respect to selecting suitable languages we compared the strategy of selecting those languages that give the best improvements when combined individually with the target language against selecting those that give the worst. Results show that selecting the best performing languages seems to be a reasonable strategy.

With respect to the question of how to select suitable languages, more experiments need to be performed. We have identified the amount of shared phonemes as a first indicator to predict the performance of the resulting system. But our work has also shown that additional metrics are required. Our goal is therefore to examine more strategies and try to find good strategies that are computationally in-expensive.

7. REFERENCES

[1] Sebastian Stüker, Markus Müller, Quoc Bao Nguyen, and Alex Waibel, "Training time reduction and performance improvements from multilingual techniques on the babel ASR task," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014.

[2] Quoc Bao Nguyen, Jonas Gehring, Markus Müller, Sebastian Stüker, and Alex Waibel, "Multilingual shift-

ing deep bottleneck features for low-resource ASR," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5607–5611.

[3] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, July 2006.

[4] Dong Yu and Michael L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *INTERSPEECH*, 2011, pp. 237–240.

[5] L. Mangu, Hong-Kwang Kuo, S. Chu, B. Kingsbury, G. Saon, Hagen Soltau, and F. Biadsy, "The IBM 2011 GALE Arabic Speech Transcription System," in *Proceedings of the ASRU*, Waikoloa, HI, USA, December 2011.

[6] T.N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-Encoder Bottleneck Features Using Deep Belief Networks," in *Proceedings of the ICASSP*, Kyoto, Japan, March 2012.

[7] Jonas Gehring, Yajie Miao, Florian Metze, and Alex Waibel, "Extracting Deep Bottleneck Features Using Stacked Auto-Encoders," in *Proceedings of the ICASSP*, Vancouver, Canada, May 2013.

[8] Pawel Swietojanski, Arnab Ghoshal, and Steve Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proceedings of the Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 246–251, IEEE.

[9] Ngoc Thang Vu, Wojtek Breiter, Florian Metze, and Tanja Schultz, "Initialization Schemes for Multilayer Perceptron Training and their Impact on ASR Performance using Multilingual Data," in *Proceedings of the INTERSPEECH*, Portland, Oregon, September 2012.

[10] T. Schultz and A. Waibel, "Language Independent and Language Adaptive Acoustic Modeling for Speech Recognition," *Speech Communication*, vol. 35, no. 1-2, pp. 31–51, August 2001.

[11] Karel Vesely, Martin Karafiat, Frantisek Grezl, Milos Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *Proceedings of the Spoken Language Technology Workshop (SLT), 2012 IEEE*. 2012, pp. 336–341, IEEE.

[12] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual Acoustic Models Using Distributed Deep Neural Networks," in *Proceedings of the ICASSP*, Vancouver, Canada, May 2013.

[13] Stefano Scanzio, Pietro Laface, Luciano Fissore, Roberto Gemello, and Franco Mana, "On the use of a multilingual neural network front-end," in *Proceedings of the Interspeech*, 2008, pp. 2711–2714.

[14] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual training of Deep-Neural networks," in *Proceedings of the ICASSP*, Vancouver, Canada, 2013.

[15] Samuel Thomas, Michael L Seltzer, Kenneth Church, and Hynek Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6704–6708.

[16] KM Knill, Mark JF Gales, Shakti P Rath, Philip C Woodland, Chao Zhang, and S-X Zhang, "Investigation of multilingual deep neural networks for spoken term detection," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 138–143.

[17] Frantisek Grézl and Martin Karafiát, "Adapting multilingual neural network hierarchy to a new language," in *Spoken Language Technologies For Under-resourced Languages (SLTU), 2014 4th International Workshop on*, 2014, pp. 39–45.

[18] Kornel Laskowski, Mattias Heldner, and Jens Edlund, "The Fundamental Frequency Variation Spectrum," in *Proceedings of the 21st Swedish Phonetics Conference (Fonetik 2008)*, Gothenburg, Sweden, June 2008, pp. 29–32.

[19] Kjell Schubert, "Grundfrequenzverfolgung und deren Anwendung in der Spracherkennung," M.S. thesis, Universität Karlsruhe (TH), Germany, 1999, In German.

[20] František Grézl, Martin Karafiát, Stanislav Kontár, and J Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proceedings of the Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, pp. V–757 – IV–760.

[21] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme recognition using time-delay neural networks," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328–339, Mar 1989.

[22] Monika Woszczyna, N. Aoki-Waibel, Finn Dag Buø, Noah Coccaro, Keiko Horiguchi, Thomas Kemp, Alon Lavie, Arthur McNair, Thomas Polzin, Ivica Rogina, Carolyn Rose, Tanja Schultz, Bernhard Suhm, M. Tomita, and Alex Waibel, "Janus 93: Towards spontaneous speech translation," in *International Conference*

on Acoustics, Speech, and Signal Processing 1994, Adelaide, Australia, 1994.

[23] Hagen Soltau, Florian Metze, Christian Fugen, and Alex Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*. IEEE, 2001, pp. 214–217.

[24] David RH Miller, Michael Kleber, Chia-Lin Kao, Owen Kimball, Thomas Colthurst, Stephen A Lowe, Richard M Schwartz, and Herbert Gish, "Rapid and accurate spoken term detection.," in *INTERSPEECH*, 2007, pp. 314–317.

The NAIST-NTT TED Talk Treebank

Graham Neubig[*], *Katsuhito Sudoh*[†], *Yusuke Oda*[*], *Kevin Duh*[*], *Hajime Tsukuda*[†], *Masaaki Nagata*[†]

[*] Nara Institute of Science and Technology, Nara, Japan
[†] NTT Communication Science Laboratories, Kyoto, Japan
neubig@is.naist.jp

Abstract

Syntactic parsing is a fundamental natural language processing technology that has proven useful in machine translation, language modeling, sentence segmentation, and a number of other applications related to speech translation. However, there is a paucity of manually annotated syntactic parsing resources for speech, and particularly for the lecture speech that is the current target of the IWSLT translation campaign. In this work, we present a new manually annotated treebank of TED talks that we hope will prove useful for investigation into the interaction between syntax and these speech-related applications. The first version of the corpus includes 1,217 sentences and 23,158 words manually annotated with parse trees, and aligned with translations in 26-43 different languages. In this paper we describe the collection of the corpus, and an analysis of its various characteristics.

1. Introduction

Syntactic parsing is widely considered as a useful component of natural language processing systems, not the least of which being machine translation [1, 2]. While a large part of the work on these applications has focused on the written word, we can assume that the fundamental principles behind syntax's success in these applications will also carry over to spoken language as well.

The great majority of recent work on syntactic parsing has been based on the statistical paradigm, in which the parameters of the parser are estimated from treebanks of manually annotated parse trees. In English, the standard data set for estimating these parsers is the Wall Street Journal section of the Penn Treebank [3], consisting of written language from newspapers. However, as there are large differences between written language and spoken language, there have also been some efforts to create resources for spoken language, including the Penn Treebank annotations of ATIS travel conversation and Switchboard telephone conversation data, as well as the OntoNotes [4] annotation of broadcast news and commentary. While these corpora mainly focus on informal speech or news, spoken monologue in the form of talks presented to an audience is also an attractive target for speech processing applications. In particular, the talk data from TED[1] has been used as a target for much research, most notably the IWSLT evaluation campaigns [5].

In this work, we present the *NAIST-NTT TED Talk Treebank*, a new manually annotated treebank of TED talks that we hope will prove useful for investigation into the interaction between syntax and speech-related applications such as speech translation. The first version of the corpus consists of a total of 10 talks, consisting of approximately 125 minutes of audio amounting to 1217 sentences. All sentences are manually annotated with parse trees following the standard Penn Treebank format. To allow for examination of the interaction between syntax and speech, all sentences are automatically time aligned with the corresponding speech file. In addition, to allow for multi-lingual research, we collected and sentence-aligned TED subtitles in anywhere from 26 to 43 languages per talk, with a total of 18 languages having translations for every talk.

In this paper, we present the details of how we constructed the corpus, including data collection, treebank annotation, speech time alignment, and multilingual sentence alignment. We also provide an analysis of the corpus, including its various characteristics and to what extent they differ from existing speech and text corpora, as well as the accuracy of an existing syntactic parser on the corpus. The corpus has been made publicly available for download under the Creative Commons License at
http://ahclab.naist.jp/resource/tedtreebank

2. Corpus Data

In this section, we describe the data used as material for the corpus.

2.1. English Data

Table 1: *Details of the annotated data.*

Set	Talk	Min.	Sent.	Word
All	10	125.07	1,217	23,158
Train	7	87.23	822	16,063
Test	3	37.84	395	7,095

The English text and speech data were gathered from TED Talks. Specifically, we gathered data starting with the beginning of

[1] http://www.ted.com

Proceedings of the 11th International Workshop on Spoken Language Translation
Lake Tahoe, December 4th and 5th, 2014

the May 2012 version of the WIT3 [6] training corpus for English-Japanese. From this data, for the first version of the treebank we chose 10 talks, the details of which are shown in Table 1.[2]

As the original TED data is subtitles, it is necessary to group these subtitles into sentences before performing annotation. In the creation of the corpus, we used the standard English sentence segmentation provided by the WIT3 data.[3]

In addition, when using a corpus for experiments, it is desirable to have a "standard" split between the training and testing data. As this standard, we designated a split of the first 7 talks as training data, and the other 3 talks as test data, resulting in an approximately 2/3 of the corpus for training, and 1/3 for testing when counting the number of sentences. This is also the split used in the analysis in Section 5.

With regards to the characteristics of the speeches and the speakers, the collected data is, like TED as a whole, quite diverse. Of the ten talks, 9 have a single speaker, and 1 has two speakers. Of these 11 speakers, 7 are men, and 4 are women.

2.2. Multilingual Data

In addition, because most of the talks in the collection have been translated into several other languages, we also downloaded the subtitles for all other languages in which they existed. As a result, for each talk we obtained subtitles in 26-43 different languages. For a total of 18 languages (shown in Table 2), this resulted in subtitles for all the parsed talks, and for 37 languages there were subtitles for some, but not all of the talks. We further combined these subtitles together into units that correspond to each English sentence, creating a sentence-aligned corpus between all of the languages.[4]

Table 2: *Languages for which subtitles existed for all 10 annotated talks.*

Arabic, Bulgarian, German, Greek, Spanish, French, Hebrew, Italian, Japanese, Korean, Dutch, Polish, Brazilian Portuguese, Romanian, Russian, Turkish, Simplified Chinese, Traditional Chinese

While there exist other corpora of sentence-aligned TED talks [6], and other corpora of bilingually aligned syntax trees [7], to our knowledge this is the first corpus with manually annotated syntax trees in English and translations into a large number of languages, and also the first multilingually aligned treebank of the spoken word. We hope that this data will be of use for investigations into the effect of syntax on speech translation and other cross-lingual tasks.

3. Creation of Parse Trees

The first, and most labor-intensive annotation task was the creation of manual parse trees for the English sentences.

[2]We are currently in the process or annotating more data, which will be released as a second version of the corpus on completion.

[3]This segmentation standard groups multiple subtitles into single sentences, but never splits subtitles. Thus there are rare cases where a subtitle containing multiple sentences results in unsegmented sentences in the data.

[4]Of course, there are also a few cases where a single English sentence corresponds to multiple sentences, or less than one sentence in the foreign language.

3.1. Annotation Standard

The most important part of creating a treebank is coming up with an appropriate annotation standard. Fortunately, the extensive 318-page annotation standard for the Penn Treebank exists,[5] and we choose to adopt this standard to maintain intercompatibility with the Penn Treebank. Specifically, we follow the actual documentation of the Treebank II annotation standard, but only annotate constituent labels (e.g. "NP"), omitting tagging of syntactic roles (e.g. the "-SUBJ" in "NP-SUBJ") or null elements (e.g. the omitted subject due to wh-movement in questions). We chose this annotation standard because most treebank parsers, such as the Berkeley parser, are trained on and generate annotation without constituent labels or null elements.

We also make one minor modification of the treebank standard tailored to the speech that appears in TED. Specifically, within TED talks, there are many cases in which the speaker quotes the words of another. The quote annotation in the Penn Treebank, in contrast to the annotation of other phenomena such as parenthesized expressions, simply treats each element of a quote as elements of its surrounding clause. In order to make the boundaries of quotes more explicit and easy to recognize, we add a single node with the symbol "QUOTE" showing the boundaries of a quote, as is done for parenthesized expressions. It should be noted that this change is automatically reversible, and the Penn Treebank annotation can be completely recovered by simply removing the QUOTE node and promoting its children.

An example of an annotated tree, including a QUOTE annotation is shown in Figure 1.

3.2. Annotation Process

Treebank annotation is an extremely time consuming process, particularly when the entirety of the tree has to be created from scratch. Fortunately, relatively accurate treebank parsers already exist, allowing us to create an initial parse first using an off-the-shelf parser, then have annotators spend their time fixing the errors of the existing parser. In this case, we use the Berkeley Parser[6] [8] to create an initial parse.

After this, we hired annotators to go through the trees and annotated them based on the standard described in the previous section. The annotators are well versed in annotation of linguistic data, and were given the standard and asked to follow it closely. After receiving this initial annotation result, the first author of the paper went through the entirety of the corpus, checking once more for any remaining errors. Finally, the trees were automatically checked for inconsistencies such as duplicated unary rules, or trees that were judged as a warning or error according to the phrase structure conversion tools of Johansson and Nugues [9].

4. Speech Time Alignment

Because the treebank described in this paper is of spoken language, the correspondence between syntactic trees and features of the speech is of particular interest. For example, it has been previously noted that prosody and syntax have a close relationship [10], and this corpus could be used to perform further investigations into these and other issues.

In order to create the time alignment of each word in the speech,

[5]`http://www.cis.upenn.edu/~treebank/`

[6]`https://code.google.com/p/berkeleyparser/`

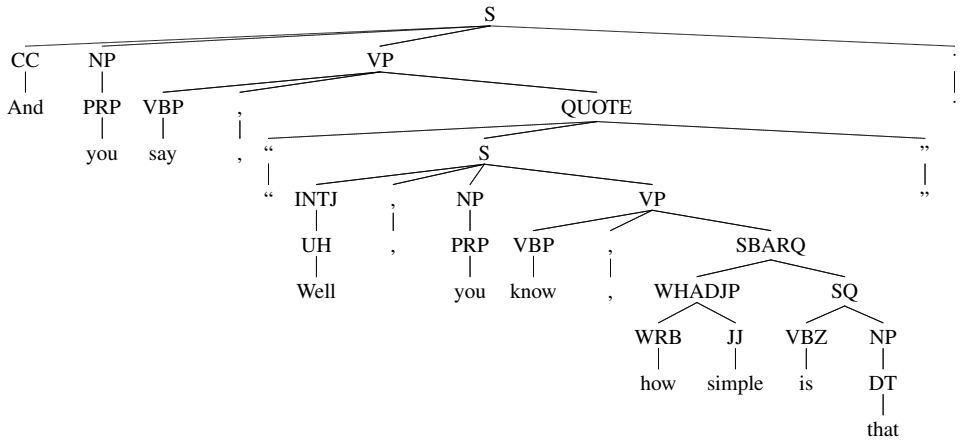


Figure 1: *An example tree from TED including QUOTE annotation.*

we prepared the data according to an automatic process. In the first step of the process, we performed forced decoding using the Kaldi decoder [11] with a model trained for the IWSLT speech recognition task [12]. In addition, as there are small differences between the transcripts used in forced decoding and the actual subtitles, due to factors such as punctuation deletion and normalization of numbers, we further aligned the times found in the forced alignment to the words in the subtitles, which were used in the annotation of the parse trees.

5. Analysis

In this section, we describe our analysis of the prepared corpus, first listing statistics of the trees in the corpus, measuring parsing accuracy and analyzing parsing errors.

5.1. Corpus Statistics

First, in this section we describe statistics of the collected parse trees for TED in comparison to the Wall Street Journal (WSJ) section of the Penn Treebank and the Broadcast News (BN) and Broadcast Commentary (BC) sections of OntoNotes. In particular, we focus on the differences in complexity of the sentences, as well as the different types of syntactic structures that appear in the sentences.

5.1.1. Syntactic Complexity

The first and most simple statistic that comes to mind regarding the complexity of the sentences is sentence length. In Figure 2 we show a histogram of the sentence lengths for the two corpora (after tokenization). From this figure we can see, perhaps as expected, that there is a larger number of long sentences in the newspaper text of WSJ. However, there are still a significant number of long sentences in TED with approximately 40% of sentences being 20 words or more. Compared with the two corpora of broadcast news and commentary, we can see that the length characteristics of the corpus are quite similar to those of broadcast news, and significantly longer than the more spontaneous broadcast commentary.

In addition to the length, it is also possible to examine the syntactic trees directly to understand the syntactic complexity of the sentences. There are a number of measures of syntactic complexity, and according to Roark et al. [13], who examine the correlation

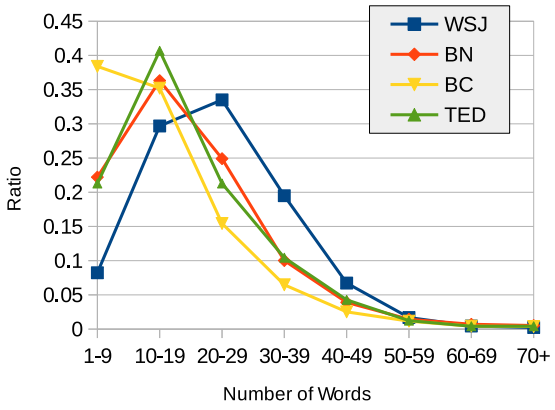


Figure 2: *A histogram of sentence lengths in Wall Street Journal (WSJ), Broadcast News (BN), Broadcast Commentary (BC), and TED.*

of several syntactic complexity measures with neuropsychological tests, two measures show a significant correlation with psychological factors such as the burden on memory. The first is simply the ratio of internal tree nodes to words in the sentence. The second is Frazier's measure of syntactic complexity [14], which is inspired by the number of syntactic elements that must be held in working memory. Specifically, it is defined as the average distance between a terminal node in the syntactic tree and its first ancestor that is not a leftmost sibling, with sentence nodes counting 1.5 times as much as other nodes (more details can be found in the referenced paper).

Table 3: *Syntactic complexity for sentences of length 10-29.*

Measure	WSJ	BN	BC	TED
Frazier	0.766	0.836	0.884	0.832
Nodes/Word	2.781	2.855	2.897	2.874

In Table 3 we show the values of these two complexity mea-

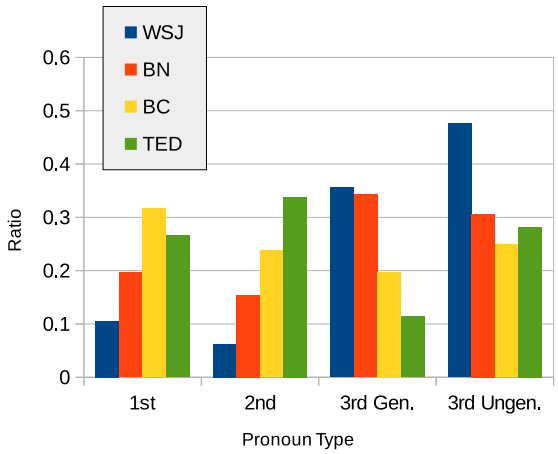


Figure 3: *The distribution of pronoun types for each corpus.*

sures for the 4 corpora under consideration, limiting our analysis to sentences of length 10-29 to reduce any artificial effects of analyzing different length sentences. From the results, we can see that WSJ has the lowest scores, BC has the highest scores, and TED and BN are relatively similar. While it seems somewhat counterintuitive that the more conversational corpora have more syntactic complexity, in fact news text is carefully planned and edited, often resulting in sentences that are easier to interpret than those in more informal speech.

5.1.2. Stylistic Difference

As the previous statistics show that the complexity of sentences in the TED corpus are similar to those of broadcast news, it is of interest whether there are stylistic differences that set it apart. It is somewhat difficult to pick apart stylistic differences quantitatively as simple statistics such as unigram distributions conflate stylistic and topical differences, so we calculated a variety of statistics and here focus on two simple statistics in which TED stood out.

First, in Figure 3, we show the difference in the distribution of singular pronouns, grouped into the first person (I/me), second person (you), third person gendered (he/she/him/her), and third person ungendered (it). From this figure, we can see that TED is unique in having more second person pronouns than any other category, demonstrating how TED speakers attempt to reference and engage their audience. In this way, the corpus is most similar to BC, which also contains a large number of 1st and 2nd person references, and in stark contrast to news, for which the large majority of pronouns are in the 3rd person.

Table 4: *Percentage of present, past, and progressive verbs.*

Tense	WSJ	BN	BC	TED
Present	42.8	50.2	56.1	64.0
Past	38.4	29.7	27.7	18.7
Prog.	18.7	20.1	16.1	17.3

Second, in Table 4, we show statistics about the tense of verbs, whether in the present (VBP/VBZ), past (VBD), or progressive (VBG) tense. From this table, we can see that as we move from news to conversation to TED, the number of past tense verbs decreases, and the number of present tense verbs increases. This marks a notable difference between news, which often looks backwards on the past, and the TED talks, which are often focused on what the speaker is doing now, or looking forward into the future.

In summary of the analysis, TED represents broadcast news in sentence complexity, but is also close to broadcast conversation in two stylistic characteristics. Thus, TED is somewhat different from these other genres, and thus manually annotated syntactic resources for TED are likely to give a benefit in the processing of TED talks and other similar monologues. In the following section, we examine this further in parsing experiments using the TED treebank.

5.2. Parsing Experiments

In order to test the accuracy of automatic parsing over the TED treebank, we performed parsing experiments, comparing with the WSJ section of the Penn Treebank.

5.2.1. Experimental Setting and Accuracy

We used two different sets of training data. The *wsj-train* data includes WSJ sections 2 to 21, which is the standard setting for training parsers on the Penn Treebank. The *wsj+ted-train* data also includes TED treebank training data (the first 7 talks, as specified in Section 2.1) in addition to *wsj-train*. We also prepared two data sets for testing each model. The *wsj-test* data includes WSJ section 23, the standard testing setting for evaluating syntactic parsers on WSJ. and *ted-test* data includes the TED treebank testing data (again specified in Section 2.1 as the last 3 talks). All "QUOTE" tags in the TED treebank are removed before training and testing, in order to ensure consistency with WSJ.

The Berkeley Parser [8] is used to train a latent annotated probabilistic context free grammar (PCFGLA) model from each of the training data sets and to generate a one-best parse of test data using trained model. We used EVALB[7] to evaluate parsing accuracy of each result in the form of bracketing F1 measure.

Table 5 shows the bracketing F1 measure for test sentences that have 40 words or less in each train/test data combination. Numbers in bold indicate the model with the better accuracy using the same test data. From these results, we can see that on the *ted-test* data, the model trained using the *wsj+ted-train* data achieves somewhat better performance than the model trained with only *wsj-train*. For *wsj-test*, the difference is slim, with both models achieving largely the same accuracy.

These results indicate that just by adding a small number of TED sentences to the WSJ data for training, we are able to achieve a small gain in parsing accuracy on the TED data. It should be noted that this is the simplest possible method for domain adaptation, and it is likely that there is still significant room for improvement by using more sophisticated techniques to account for the fact that the TED data is still significantly smaller than the WSJ data.

5.2.2. Individual Examples

Figures 4 and 5 show examples of parse trees of a sentence from the test set[8] trained with the *wsj-train* model, and the *wsj+ted* model respectively. The correct parse is the same as that generated by the *wsj+ted* model, with the exception that "(NN soap)" should be "(NP

[7] `http://nlp.cs.nyu.edu/evalb/`

[8] The example was actually slightly shortened by removing two elements from the long coordinate phrase to ensure that it fit on one page.

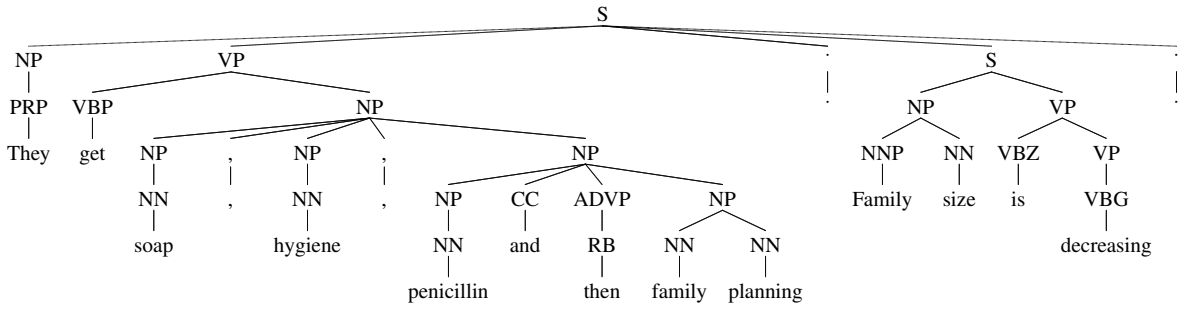


Figure 4: *Example of best parse using the* wsj-train *grammar.*

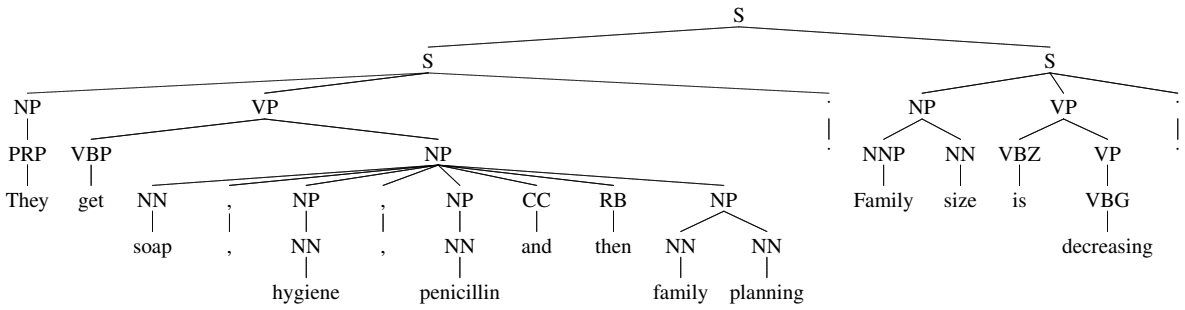


Figure 5: *Example of the best parse using the* wsj+ted-train *grammar.*

Table 5: *Bracketing F1 measure of each parsing evaluation.*

		Train	
		wsj-train	*wsj+ted-train*
Test	*wsj-test*	**90.41**	90.38
	ted-test	88.65	**88.99**

(NN soap))," and "(NNP Family)" should be "(NN Family)." This sentence has two notable characteristics.

First, there are multiple sentences in one tree, because TED data is based on subtitles of actual talks. These multiple sentence lines often occur when multiple sentences are included within a single subtitle. This is in contrast to the WSJ in which each line is split at sentence boundaries before annotation. As a result, the model trained using only the WSJ corpus tends to misparse lines including multiple sentences as single sentence.[9] On the other hand, model trained including the TED treebank expresses them using the (S → S S) rule and can parse sentences with this characteristic properly, although it does still make the mistake of determining that "Family" is a proper noun.

Second, the model trained by *wsj+ted-train* data makes a better parse of the long parallel noun phrase. In this example, the words "penicillin and then family planning" should be immediate children of the parent NP as in Figure 5, not an independent phrase as in Figure 4.

[9] A WSJ treebank parser with an extra sentence segmentation preprocessing step could also likely parse this example properly, but it this does add additional complexity that can be largely avoided by training a model that can handle these lines properly.

6. Conclusions

In this paper, we presented a treebank consisting of material from TED talks, an example of spoken language monologue sparsely covered by existing resources. The corpus consists of manually annotated syntactic trees, corresponding speech, time alignments, and multilingual translations. We hope that this corpus will be of use for examining the interaction between syntax and speech translation, or other applications of NLP to speech.

As future work, we are currently continuing annotation of the corpus, and plan to release an expanded second version of the corpus upon completion of this annotation. We also plan on performing more comprehensive parsing experiments using domain adaptation techniques, and examining the effect of parsing on the accuracy on machine translation.

7. References

[1] K. Yamada and K. Knight, "A syntax-based statistical translation model," in *Proc. ACL*, 2001.

[2] G. Neubig and K. Duh, "On the elements of an accurate tree-to-string machine translation system," in *Proc. ACL*, 2014.

[3] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The Penn treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[4] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "OntoNotes: the 90% solution," in *Proc. HLT*, 2006, pp. 57–60.

[5] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. IWSLT*, 2012.

[6] M. Cettolo, C. Girardi, and M. Federico, "WIT3: web inventory of transcribed and translated talks," 2012, pp. 261–268.

[7] N. Xue, F. Xia, F.-D. Chiou, and M. Palmer, "The Penn Chinese treebank: Phrase structure annotation of a large corpus," *Natural language engineering*, vol. 11, no. 02, pp. 207–238, 2005.

[8] S. Petrov, L. Barrett, R. Thibaux, and D. Klein, "Learning accurate, compact, and interpretable tree annotation," in *Proc. ACL*, 2006, pp. 433–440.

[9] R. Johansson and P. Nugues, "Extended constituent-to-dependency conversion for english," in *16th Nordic Conference of Computational Linguistics*, 2007, pp. 105–112.

[10] S.-A. Jun, "Prosodic phrasing and attachment preferences," *Journal of Psycholinguistic Research*, vol. 32, no. 2, pp. 219–249, 2003.

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1–4.

[12] S. Sakti, K. Kubo, G. Neubig, T. Toda, and S. Nakamura, "The NAIST english speech recognition system for IWSLT 2013," in *Proc. IWSLT*, 2013.

[13] B. Roark, M. Mitchell, and K. Hollingshead, "Syntactic complexity measures for detecting mild cognitive impairment," in *Proc. BioNLP*, 2007, pp. 1–8.

[14] L. Frazier, "Syntactic complexity," *Natural language parsing: Psychological, computational, and theoretical perspectives*, pp. 129–189, 1985.

Better Punctuation Prediction with Hierarchical Phrase-Based Translation

Stephan Peitz, Markus Freitag, Hermann Ney

Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
<surname>@cs.rwth-aachen.de

Abstract

Punctuation prediction is an important task in spoken language translation and can be performed by using a monolingual phrase-based translation system to translate from unpunctuated to text with punctuation. However, a punctuation prediction system based on phrase-based translation is not able to capture long-range dependencies between words and punctuation marks. In this paper, we propose to employ hierarchical translation in place of phrase-based translation and show that this approach is more robust for unseen word sequences. Furthermore, we analyze different optimization criteria for tuning the scaling factors of a monolingual statistical machine translation system. In our experiments, we compare the new approach with other punctuation prediction methods and show improvements in terms of F_1-Score and BLEU on the IWSLT 2014 German→English and English→French translation tasks.

1. Introduction

Spoken language translation (SLT) has become an important application of automatic speech recognition (ASR) and machine translation (MT). The challenge of SLT is to translate automatic transcribed speech rather than written text into another language. In recent years, several research projects such as QUAERO[1] and EU-Bridge[2] have been focussed on speech translation. Furthmore, the increasing number of available Android application for speech translation[3] indicates a growing interest in speech translation technologies in the general public.

The translation of speech is in general divided in two independent parts. First, ASR provides a automatic transcription of spoken words. Next, the recognized words are translated by an MT system.

As in speech punctuation is not made explicit, most ASR systems provide an output without punctuation marks only. Most MT systems however are trained on data with proper punctuation and expect written text with correct punctuation as input. Therefore, the output of ASR systems has to be enriched with punctuation marks. In MT an accurate punctuation of the input is crucial as the prediction errors affect the translation quality. In [1], a loss of up to 4 BLEU points was obtained if punctuation marks need to be predicted, compared to correct punctuation in the input.

In recent years several methods to predict punctuation were developed. These methods are based on n-gram language models, on conditional random fields (CRF) or on monolingual statistical machine translation (SMT) systems translating from unpunctuated text to text with punctuation. One of the advantages of an SMT system or CRF is that more features beside the language model can be integrated. Furthermore, punctuation prediction can be done before, after or during the actual translation. Following [1, 2], we use a phrase-based SMT system for punctuation prediction before the actual translation as starting point.

In this work, we propose to employ hierarchical translation in place of phrase-based translation. In phrase-based translation, the translation units are bilingual phrases which are pairs formed by a sequence of source language words and its translation. Since a sequence of words can be translated at once, local contextual information is preserved. In the context of punctuation prediction, such information is useful to predict punctuation marks depending of its surrounding words, e.g. commas. However, this approach has its limitation for unseen word sequences and dependencies beyond the local context, e.g. the dependency between a question word and a question mark. If a sequence of words was not seen in the training data, the phrase-based translation system will fall back on shorter phrases with less local contextual information. Thus, more prediction errors can occur. To generalize better and to model dependencies as described above, we need a more abstract form of phrases. In hierarchical translation, such phrases are defined since discontinuous phrases with "gaps" are allowed. Those phrases capture long-range dependencies between words. In terms of punctuation prediction, we want to model dependencies between words and punctuation marks. In addition, by using more abstract phrases, a punctuation prediction system based on hi-

[1] http://www.quaero.org/
[2] http://www.eu-bridge.eu/
[3] https://play.google.com/store/search?q=speech%20translation&c=apps

erarchical translation models is more robust for unseen word sequences and generalize better.

As already mentioned, another advantage of using an SMT system for punctuation prediction is that different features besides the language model can be applied. These features are combined in a log-linear model. In this work, we investigate the impact of different optimization criteria for tuning the scaling factors of the features with minimum error rate training.

In our experiments on the IWSLT 2014 German→English and English→French machine translation task, we show improvements in terms of F_1-Score and BLEU.

This paper is structured as follows. We start in Section 2 with a short overview of the published research on punctuation prediction. In Section 3, we recap the idea of modeling punctuation prediction as machine translation and discuss different optimization criteria for tuning the scaling factors of a monolingual MT system. We present our approach using a hierarchical phrase-based translation system for punctuation prediction in Section 4. Finally, Section 5 describes the experimental results, followed by a conclusion in Section 6.

2. Related Work

In recent years, several approaches for predicting punctuation have been presented.

The HIDDEN-NGRAM tool from the SRI toolkit [3] considers punctuation marks as hidden events occurring between words. The most likely hidden tag sequence is found using an n-gram language model trained on punctuated text. In this work, we will compare with this tool.

The approach described in [4] is based on conditional random fields. They extended the linear-chain CRF model to a factorial CRF model using two layers with different sets of tags for punctuation marks respectively sentence types. They compared their approach with linear-chain CRF model and the HIDDEN-NGRAM tool on the IWSLT 2009 corpus. Besides the comparison of the translation quality in terms of BLEU, they also compared the CRF models with the hidden event language model regarding precision, recall and F_1-Score. Both in terms of BLEU and in terms of precision, recall and F_1-Score the CRF models outperformed the hidden event language model. They claimed that using non-independent and overlapping features of the discriminative model as machine translation instead of a language model only helped. Similar to this approach, using a statistical machine translation system for punctuation prediction has the advantage to integrate more features beside the language model.

Using MT for punctuation prediction was first described in [5]. In this work, a phrase-based statistical machine translation system was trained on a pseudo-bilingual corpus. The case-sensitive target language text with punctuation was considered as the target language and the text without case information and punctuation was used as source language. They applied this approach as postprocessing step in evaluation campaign of IWSLT 2007 and achieved a significant improvement over the baseline. In [6] the same approach was employed as preprocessing step and compared with the HIDDEN-NGRAM tool within the evaluation campaign of IWSLT 2008. The HIDDEN-NGRAM tool outperformed the MT-based punctuation prediction. In addition to punctuation prediction using a monolingual MT system, performing segmentation of ASR output was described in [2]. In all mentioned papers using a monolingual MT system for punctuation prediction, the optimization criterion for tuning the scaling factors of such a system was not described. In this work, we will tune both the phrase-based and the hierarchical translation system against BLEU and F_α-Score and analyze the impact on the prediction accuracy and translation quality.

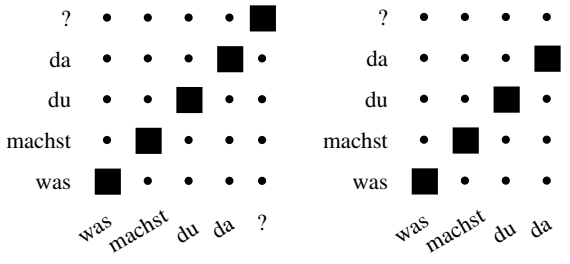
In [7], three different stages at which punctuation can be predicted are investigated: before, during and after the translation. Each of the stages requires a different translation system and has advantages and disadvantages. For predicting punctuation during the translation, additional punctuation prediction is not needed. The punctuation prediction before and after the translation was done with the HIDDEN-NGRAM tool. The implicit punctuation generation worked best on IWSLT 2006 corpus, but on TC-STAR 2006 corpus they achieved better results with punctuation prediction before and after the actual translation.

The impact of using a monolingual statistical machine translation system rather than the HIDDEN-NGRAM tool was analyzed in [1]. The authors report an improvement of 0.8 BLEU points by applying a monolingual statistical machine translation system before translation. An important advantage is that no modification of the actual translation system is needed. In our work, we follow this pipeline and replace the phrase-based translation model by a hierarchical translation model.

3. Modeling Punctuation Prediction as Machine Translation

Punctuation prediction using a statistical machine translation system is based on following pipeline. First, we extract the translation model for the SMT system from a pseudo-bilingual corpus. In order to create such a corpus, we need two versions of a monolingual corpus: one without punctuation (source text) and one with punctuation (target text). This is done by creating a monotone alignment (Figure 1(a)) and removing punctuation marks from the source sentences. The punctuation marks in the target sentences which are aligned with punctuation marks in the source sentences become unaligned (Figure 1(b)).

Given the pseudo-bilingual corpus and the modified alignment, we extract the translation model. In our work, we substitute the phrase-based translation model by a hierarchical translation model. Details about hierarchical translation are given in Section 4. In a next step, the scaling factors of



(a) Monotone alignment

(b) Punctuation marks in the target sentence become unaligned

Figure 1: Modification of the alignment

the monolingual translation system are tuned. We get a tuning set by removing the punctuation marks from a development set and use the original development set as reference. In this paper, we analyze different criteria used in the optimization of the scaling factors. We give further details in the following subsection.

3.1. Optimization Criteria

In most state-of-the-art SMT systems, MERT [8] is applied to optimize scaling factors of features using BLEU [9] as optimization criterion. However, the performances of systems predicting punctuation are measured and compared with the F_1-Score which is the harmonic mean of precision and recall. Thus, there is an inconsistency between optimization criterion and metric. Furthermore, the F_1-Score considers both precision and recall while BLEU is a metric which is based on n-gram precision and does not take recall into account. A criterion including recall is important because it ensures that the punctuation prediction system generates an appropriate amount of punctuation marks. In this work, we use F_α-Score as a more suitable optimization criterion. F_α-Score is a more general form of the F_1-Score, where α is a positive real number:

$$F_\alpha = (1+\alpha) \cdot \frac{(precision \cdot recall)}{\alpha \cdot precision + recall}$$

By varying the parameter α, more emphasis can be put on recall or precision. In this work, we will put more weight on recall and tune the systems with $\alpha = \{1,2,3,4\}$. We might lose precision and overgenerate punctuation marks, but this could be compensable for the actual translation system.

However, tuning a system on F_α-Score directly would not be practical as the positions of the punctuation marks would be ignored. For the optimization, we have to modify the F_α-Score and take the predecessors of the punctuation marks into account. In this work, we tune our monolingual translation systems using the modified F_α-Score as criterion with $\alpha = \{1,2,3,4\}$ and compare against systems tuned on BLEU.

4. Punctuation Prediction based on Hierarchical Translation

In hierarchical phrase-based translation [10], discontinuous phrases with "gaps" are allowed. The translation model is formalized as a synchronous context-free grammar (SCFG) and consists of bilingual rules, which are based on bilingual standard phrases and discontinuous phrases. Each bilingual rule rewrites a generic non-terminal X into a pair of strings $\tilde{f}$ and $\tilde{e}$ with both terminals and non-terminals in both languages

$$X \rightarrow \langle \tilde{f}, \tilde{e} \rangle.$$

Obtaining these rules is based on a heuristic extraction from automatically word-aligned bilingual training data. Just like in the phrase-based approach, all bilingual rules of a sentence pair are extracted given an alignment. The standard phrases are stored as *lexical rules* in the rule set. In addition, whenever a phrase contains a sub-phrase, this subphrase is replaced by a generic non-terminal X. With these hierarchical phrases we can define the *hierarchical rules* in the SCFG. The rule probabilities which are in general defined as relative frequencies are computed based on the joint counts $C(X \rightarrow \langle \tilde{f}, \tilde{e} \rangle)$ of a bilingual rule $X \rightarrow \langle \tilde{f}, \tilde{e} \rangle$

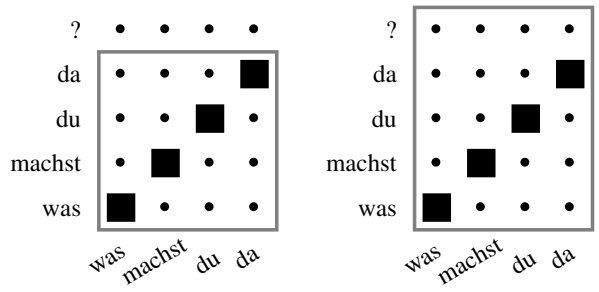
$$p(\tilde{f}|\tilde{e}) = \frac{C(X \rightarrow \langle \tilde{f}, \tilde{e} \rangle)}{\sum_{\tilde{f}'} C(X \rightarrow \langle \tilde{f}', \tilde{e} \rangle)}.$$

The translation probabilities are computed in source-to-target as well as in target-to-source direction. In the translation processes, these probabilities are integrated in the log-linear combination among other models such as a language model, word lexicon models, word and phrase penalty and binary features marking hierarchical phrases, glue rule and rules with non-terminals at the boundaries.

The translation process of hierarchical phrase-based approach can be considered as a parsing problem. Given an input sentence in the source language, this sentence is parsed using the source language part of the SCFG. Using the associated target part of the applied rule, a translation can be constructed. The language model score is incorporated by employing the cube pruning algorithm presented in [11].

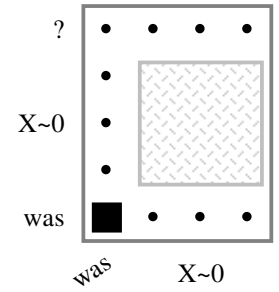
In a standard translation task, hierarchical rules with up to two non-terminals are extracted. Using rules with one non-terminal, the translation system is able to model long-range dependency between terminals. Furthermore, rules with two non-terminals make it possible to perform reordering without an additional model. In other words, the reordering is modelled in the hierarchical translation model implicitly. In case of punctuation prediction, we perform monotone translation and reordering is not necessary. Thus, we extract rules with one non-terminal maximum.

For punctuation prediction, our goal is to capture long-range dependencies between words and punctuation using hierarchical rules. To be able to extract such rules, we add an heuristic to the rule extraction process as described in the next section.



(a) Standard phrase block

(b) Extented phrase block using additional extraction heuristic



(c) Hierarchical rule extracted from extended phrase block

Figure 2: Extraction heuristic applied for initial phrase blocks

4.1. Additional Phrase Extraction Heuristic

As mentioned in the Section 3, punctuation marks in the target sentences which are aligned with punctuation marks in the source sentences become unaligned. Applying the standard phrase extraction procedure [12], phrases with punctuation are not extracted (Figure 2(a)). In order to add phrases such as

$$\langle \text{was machst du da}, \text{was machst du da ?} \rangle$$

to the translation model, we apply a heuristic which allows for phrase blocks including non-aligned words which are adjacent to phrase boundaries (Figure 2(b)). By using such additional phrases as initial phrases in the hierarchical extraction process (Figure 2(c)), we are able to extract hierarchical rules which model long-range dependencies between words and punctuation marks, e.g.

$$X \rightarrow \langle \text{was } X\tilde{}0, \text{was } X\tilde{}0 \text{ ?} \rangle,$$
$$X \rightarrow \langle \text{machst du } X\tilde{}0, \text{machst du } X\tilde{}0 \text{ ?} \rangle.$$

In the first rule, the question mark on the target side is related to the German question word "was". In the second rule, the typical German word order for questions (verb "machst" before subject "du") triggers a question mark on the target side. Both rules are more abstract since the gap could be filled with any other phrases during decoding. Even for unseen word sequences, e.g. "was machst du heute", these rules match. Thus, punctuation prediction based hierarchical translation can generalize better and improve the prediction accuracy.

In the experimental evaluation, we will analyze if such rules influence the decoding process and affect the punctuation prediction accuracy. Note, for the phrase-based translation system, we apply the non-aligned word heuristic as well.

Table 1: Data statistics for the preprocessed German-without-punctuation→German parallel in-domain training corpus used for punctuation prediction with the monolingual MT systems.

	German without Punct.	German
Sentences	171721	
Running words	2.7M	3.3M
Vocabulary	119242	119266

5. Experimental Evaluation

Our approach to use hierarchical phrase-based translation for punctuation prediction was evaluated on the IWSLT 2014 German→English and English→French machine translation tasks. IWSLT is an annual public evaluation campaign focusing on spoken language translation. The domain is lecture-type talks presented at TED conferences. The translation part of the evaluation campaign is divided into two different tracks: translation of automatic and translation of manual transcriptions. While the correct manual transcription contains punctuation marks, the automatic transcription did not. As we focus on punctuation prediction in this work, we used for the experiments *pseudo ASR output* rather than real ASR output as input. Pseudo ASR output is created by removing punctuation marks from the manual transcriptions. Thus, recognition errors do not occur and case information is preserved.

5.1. Punctuation Prediction

Both phrase-based and hierarchical translation models are trained on the provided in-domain training data (Table 1).

For tuning and testing our monolingual translation system, we used the provided manual transcribed development set and test sets (Table 2).

Training data as well as development and test set were modified as described in Section 3. A 5-gram language model, which was applied by both monolingual translation systems and the HIDDEN-NGRAM toolkit, was trained on the concatenation of the in-domain, europarl, news-commentary and commoncrawl corpora with the KenLM language model

Table 2: Data statistics for the preprocessed German-without-punctuation→German development and test sets for tuning and testing the punctuation prediction systems.

		German without Punct.	German
dev	Sentences	887	
	Running words	16521	19152
	Vocabulary	4029	4039
test	Sentences	1565	
	Running words	25483	30332
	Vocabulary	4976	4987

toolkit [13] using modified Kneser-Ney smoothing [14]. For creating the monolingual translation systems, we used an open-source translation tookit, which implements both phrase-based and hierarchical translation.

5.2. Bilingual Translation Systems

We set up translation systems for German→English and English→French to investigate the impact of better punctuation prediction on the translation quality in terms of BLEU and TER [15]. In order to analyze the effect of prediction errors on the translation quality, we compare with a setup with correct punctuation in the input. We employed phrase-based translation for both language pairs. Both systems were trained on all available bilingual and monolingual data provided by the IWSLT evaluation campaign.

5.3. Comparison of the Prediction Accuracy

The punctuation performance of our new approach using a hierarchical translation system (HPBT) is compared with a phrase-based translation system (PBT) and the HIDDEN-NGRAM tool. The accuracy is measured in precision (Prec.), recall (Rec.) and F_1-Score (F_1). Furthermore, we analyze the impact of different optimization criteria. Both translation systems were tuned on BLEU and F_α, where $\alpha = \{1,2,3,4\}$. Table 3 shows the result of this comparison for the German language.

The HPBT translation system tuned on F_2 performs best in terms of F_1-Score. For the PBT translation systems, tuning on F_2 leads to slightly better results. The performance of the HIDDEN-NGRAM toolkit is slightly better than the best PBT system. However, HIDDEN-NGRAM performs worse than the HPBT system tuned on F_2. In general, it seems tuning on F_α works better than tuning on BLEU. Although systems tuned on F_α tend to be less precise, the F_1-Score is higher compare to system tuned on BLEU. Best performance is achieved with $\alpha = 2$.

In the following, we define the PBT system tuned on BLEU as *baseline* and compare it against PBT tuned on F_2, HPBT tuned on F_2 and HIDDEN-NGRAM.

Table 3: Accuracy of the predicted punctuation on the test set of correct manual transcription without punctuation (German).

system	tuned on	Prec.	Rec.	F_1
PBT w/o heuristic	BLEU	86.7	24.4	43.9
PBT	BLEU	*82.7*	*67.5*	*74.3*
	F_1	82.6	67.5	74.3
	F_2	**78.3**	**71.4**	**74.7**
	F_3	76.6	72.2	74.4
	F_4	72.5	73.6	73.0
HPBT	BLEU	86.4	65.5	74.7
	F_1	81.8	71.0	76.0
	F_2	**77.0**	**75.4**	**76.2**
	F_3	75.9	75.2	75.6
	F_4	71.8	73.7	74.2
HIDDEN-NGRAM	-	**82.7**	**69.5**	**75.5**

Table 4: Three different classes of punctuation marks and their relative frequencies in the test set of the correct manual transcription.

Class	Punctuation marks	rel. freq.
1	. ? !	40,2%
2	,	53,3%
3	" ' ;) (	6,5%

To get further insight, on which level type of annotation the prediction methods are more or less accurate, we measure the accuracy regarding three different classes of punctuation marks (Table 4).

The result of this comparison is given in Table 5. In all three classes, HPBT outperforms both HIDDEN-NGRAM and PBT in terms of F_1-Score. However, the largest difference in accuracy is obtained in class 3.

Next, we investigate the usage of hierarchical rules in the decoding process and analyze whether such rules help to increase the prediction accuracy. This is done by counting the lexical and hierarchical rules applied during decoding. In particular, we count rules which introduce punctuation marks and compute the average target length of the applied rules. In this analysis, we compare PBT and HPBT (Table 6). While the PBT system uses short phrases with a limited local context (average target phrase length of 2.1 or 2.0), the HPBT system employs both lexical and hierarchical rules to insert punctuation marks. Even if only 20% of the rules which introduce punctuation marks are hierarchical, it seems that those rules help to improve the prediction accuracy.

We further examine these results using a prediction example (Table 7). In this example, both HIDDEN-NGRAM and PBT did not predict the question mark. However, HPBT is

Table 5: Accuracy of the predicted punctuation on the test of correct manual transcription without punctuation regarding three different classes of punctuation marks (German).

system	tuned on	class 1			class 2			class 3		
		Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
PBT	BLEU	88.4	73.9	80.5	84.8	75.9	80.1	56.7	21.3	30.9
PBT	F_2	88.7	73.8	80.5	78.5	83.5	80.9	58.8	27.1	37.1
HPBT	F_2	88.4	81.6	84.9	77.7	85.2	81.3	49.5	30.8	38.0
HIDDEN-NGRAM	-	87.8	81.6	84.6	84.8	75.1	79.7	39.9	12.6	19.1

Table 6: Comparison of the numbers of applied phrases introducing punctuation marks. Average target phrase length is given in parentheses.

system	tuned on	lexical rules	hierarchical rules
PBT	BLEU	2313 (2.1)	-
PBT	F_2	2549 (2.0)	-
HPBT	F_2	2234 (2.6)	442 (3.9)

Table 7: Examples for punctuation prediction on German pseudo ASR output using different prediction approaches.

system	tuned on	
pseudo ASR output		was machst du nur
PBT	BLEU	was machst du nur .
PBT	F_2	was machst du nur .
HPBT	F_2	was machst du nur ?
HIDDEN-NGRAM	-	was machst du nur .
correct punctuation		was machst du nur ?

able to produce the correct sentence end punctuation.

In this example, the word order "machst du" indicates that this sentence is a question. Using the HPBT system, the correct sentence end mark is introduced by applying following hierarchical rule:

$$X \rightarrow \langle \text{machst du } X\tilde{}0, \text{machst du } X\tilde{}0 \text{ ?} \rangle.$$

In this rule, a long-range dependency between the words "machst du" and the punctuation mark "?" exists. However, such a dependency is not modelled in the phrase-based translation system. The question mark can only be inserted by the phrase

$$\langle \text{nur}, \text{nur ?} \rangle.$$

A phrase with local contextual information about the word order, e.g.

$$\langle \text{machst du nur}, \text{machst du nur ?} \rangle,$$

was not seen in the extraction process and is not part of this translation model. Thus, the PBT system uses shorter phrases with less contextual information and it is more likely that a phrase producing an erroneous punctuation is used. Here, the following phrase was applied:

$$\langle \text{nur}, \text{nur .} \rangle.$$

In this example, it seems that the hierarchical system generalize better for unseen word sequences. Furthermore, the analysis shows that hierarchical rules influence the decoding process and help to improve the prediction accuracy in our experiments.

5.4. Comparison of the Translation Quality

In the introduction, we have mentioned the effect of punctuation errors on the translation quality. In the next experiments, we check whether a higher prediction accuracy results in an improvement of the translation quality in terms of BLEU and TER. We performed punctuation prediction with different setups and then translated the enriched pseudo ASR output. The translation was performed with the bilingual SMT systems described above. The result of this comparison is shown in Table 8. We lose up to 2.2 points in BLEU if punctuation marks need to be predicted. It seems that a higher prediction accuracy leads to a higher translation quality. The performance of HIDDEN-NGRAM and PBT tuned on BLEU is on the same level. By replacing the optimization criterion with F_2, we gain 0.2 points in BLEU. Using a hierarchical system improves the translation quality by additional 0.2 points in BLEU. TER is on the same level for all setups.

To verify our improvements, we carried out additional experiments on the English→French translation task (Table 9). In this setup, we performed punctuation prediction both on pseudo ASR output and real ASR output. In terms of F_1-Score, HPBT outperforms both HIDDEN-NGRAM and PBT. However, on the real ASR output test set the performance of HPBT and PBT is on a same level in terms of translation quality.

Table 8: Impact of accuracy of punctuation prediction on the translation quality (German→English). Comparison with correct punctuation in the input.

system	tuned on	Prec.	Rec.	F_1	BLEU	TER
PBT	BLEU	82.7	67.5	74.3	27.3	53.3
PBT	F_2	78.3	71.4	74.7	27.5	53.4
HPBT	F_2	**77.0**	**75.4**	**76.2**	**27.7**	**53.2**
HIDDEN-NGRAM	-	82.7	69.5	75.5	27.2	53.2
correct punctuation					29.4	51.3

Table 9: Accuracy of the predicted punctuation on the test set of automatic (ASR) and correct manual transcription without punctuation (pseudo ASR) (English→French).

					pseudo ASR		ASR	
system	tuned on	Prec.	Rec.	F_1	BLEU	TER	BLEU	TER
PBT	BLEU	81.2	67.6	73.7	28.4	54.5	22.6	62.8
PBT	F_2	72.2	75.0	73.6	28.6	55.2	22.8	63.2
HPBT	F_2	**74.8**	**77.1**	**75.9**	**28.9**	**54.7**	22.7	62.7
HIDDEN-NGRAM	-	82.0	60.2	69.4	27.0	55.4	21.7	62.6
correct punctuation					31.9	50.1	-	-

6. Conclusion

In this paper, we introduced a new approach to predict punctuation with a monolingual hierarchical translation system. While phrase-based translation is limited to local context information, we are able to model long-range dependencies between words and punctuation marks by using hierarchical translation. In our experimental evaluation, we showed that our method improves the prediction accuracy and translation quality in terms of BLEU on the IWSLT German→English and English→French translation tasks. Furthermore, tuning a monolingual translation system for predicting punctuation on F_2 rather than BLEU improves the accuracy and translation quality.

In future work, we would like go to beyond the phrase level and investigate features which are operating on sentence level. In this way, quotes or parentheses could be modelled more accurate.

7. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

8. References

[1] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling Punctuation Prediction as Machine Translation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, Dec. 2011.

[2] E. Cho, J. Niehues, and A. Waibel, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system." in *IWSLT*, 2012, pp. 252–259.

[3] A. Stolcke, "Srilman extensible language modeling toolkit," in *In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, 2002, pp. 901–904.

[4] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 177–186.

[5] H. Hassan, Y. Ma, and A. Way, "Matrex: the dcu machine translation system for iwslt 2007," in *Proceedings of the International Workshop on Spoken Language Translation 2007*, Trento, Italy, 2007.

[6] Y. Ma, J. Tinsley, H. Hassan, J. Du, and A. Way, "Exploiting Alignment Techniques in MaTrEx: the DCU Machine Translation System for IWSLT08," in *Proc. of the International Workshop on Spoken Language Translation*, Hawaii, USA, 2008, pp. 26–33.

[7] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *International Workshop on Spoken Language Translation*, Kyoto, Japan, Nov. 2006, pp. 158–165.

[8] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," Sapporo, Japan, July 2003, pp. 160–167.

[9] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, July 2002, pp. 311–318.

[10] D. Chiang, "A Hierarchical Phrase-Based Model for Statistical Machine Translation," Ann Arbor, Michigan, June 2005, pp. 263–270.

[11] Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[12] F. J. Och, C. Tillmann, H. Ney, *et al.*, "Improved alignment models for statistical machine translation," *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20–28, 1999.

[13] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable modified Kneser-Ney language model estimation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August 2013, pp. 690–696. [Online]. Available: http://kheafield.com/professional/edinburgh/estimate_paper.pdf

[14] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," Computer Science Group, Harvard University, Cambridge, Massachusetts, USA, Tech. Rep. TR-10-98, Aug. 1998.

[15] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA, August 2006, pp. 223–231.

Rule-Based Preordering on Multiple Syntactic Levels in Statistical Machine Translation

Ge Wu, Yuqi Zhang, Alexander Waibel

Institute for Anthropomatics
Karlsruhe Institute of Technology, Germany
utcur@student.kit.edu, yuqi.zhang@kit.edu, alexander.waibel@kit.edu

Abstract

We propose a novel data-driven rule-based preordering approach, which uses the tree information of multiple syntactic levels. This approach extend the tree-based reordering from one level into multiple levels, which has the capability to process more complicated reordering cases. We have conducted experiments in English-to-Chinese and Chinese-to-English translation directions. Our results show that the approach has led to improved translation quality both when it was applied separately or when it was combined with some other reordering approaches. As our reordering approach was used alone, it showed an improvement of 1.61 in BLEU score in the English-to-Chinese translation direction and an improvement of 2.16 in BLEU score in the Chinese-to-English translation direction, in comparison with the baseline, which used no word reordering. As our preordering approach were combined with the short rule [1], long rule [2] and tree rule [3] based preordering approaches, it showed further improvements of up to 0.43 in BLEU score in the English-to-Chinese translation direction and further improvements of up to 0.3 in BLEU score in the Chinese-to-English translation direction. Through the translations that used our preordering approach, we have also found many translation examples with improved syntactic structures.

1. Introduction

Word order is a general issue when we want to translate text from one language to another. Different languages normally have different word orders and the difference could be very huge. Among all the languages, Chinese is one language which is very different from English, because they belong to different language families and have long period of separately development. Both languages have a Subject-Verb-Object order, but they also have a lot of differences in word order. Especially sentences in both languages can sometimes have completely different syntactic structures. The differences may involve long-distance or unstructured position changes.

The decoder in phrase-based SMT systems is guided by language model, phrase table and reordering models, which makes word reordering possible. However, it may have some disadvantages, such as it can't handle long-distance reordering or unstructured reordering properly, or it may be rather time-consuming.

Encouraged by the results from the result from Rottmann and Vogel 2007 [1], Niehues and Kolss 2009 [2] and Herrmann et al. 2013 [3], we further propose a new data-driven, rule-based preordering method, which extracts and applies reordering rules based on syntax tree. The method is called Multi-Level-Tree (MLT) reordering, which orders the constituents on multiple levels of the syntax tree all together. This preordering method rearranges the words in source language into a similar order as they are supposed to be in the target language before translation. With the proper word order, better translation quality can be achieved. Especially, our preordering method is very suitable for translation between language pairs like English and Chinese, which have very different word orders. Besides, the method can also be combined with the above mentioned rule-based reordering methods to achieve better translation quality.

The rest of this paper is organized as follows. Section 2 presents a review of related work. In Section 3, we point out the problems for translation between English and Chinese and describe the motivation of MLT reordering. In Section 4, we introduce the details of the MLT reordering. In Section 5, we present the experimental results and evaluation. Finally, we conclude this paper in Section 6.

2. Related Work

Word reordering is an important problem for statistical machine translation, which has long been addressed.

In a phrase-based SMT system, there are several possibilities to change the word orders. Words can be reordered during the decoding phase by setting a window, which allows the decoder to choose the next word for translation. Reordering could also be influenced by the language model, because the language model gives probability of how a certain word is likely to follow. Different language model may give different probability, which further influences the decision made by log-linear model. Other ways to change the word orders include using distance based reordering models or lexicalized reordering models [4, 5]. The lexicalized reordering model

reorders the phrases by using information of how the neighboring phrases change orientations.

Another way to achieve word reordering is to detach it from decoding phase and do it separately in a pre-process before decoding, in order to reduce the time for translation. This kind of preordering approaches use linguistic information to modify the word orders. Preordering can also be rule-based, which extracts different types of reordering rules by observing reordering patterns from the training data and apply the rules to the sentences to be translated. Depends on how the rules are defined, different information may be used such as word alignments, POS tags, syntax trees, etc.

Some early approaches use manually defined reordering rules based on the linguistic information for particular languages [6, 7, 8, 9]. Later come the data-driven methods [10, 11], which learn the reordering rules automatically.

Rottmann and Vogel 2007 [1] introduced the idea of extracting reordering rules from the POS tag sequences of training data and use them for reordering. Niehues and Kolss 2009 [2] went further, and developed a method for long-distance word reordering, which works good on German-English translation task due to the long-distance shift of verbs. The method extracts discontinuous reordering rules in addition to the continuous ones, which uses a placeholder to match several words and enables the word to shift cross long distance.

Afterwards, Herrmann et al. 2013 [3] introduced a new approach to reorder the words based on syntax tree, which led to further improvements on translation quality. The algorithm takes the syntactic structure of the source sentence into account and extract the rules from the syntax tree by detecting the reordering of child sequences. It is also possible to compute reorderings only based on part of the child sequences, which is suitable for language with flat syntactic structures such as German [12].

However, these approaches which are based on POS tag sequences or syntax trees may not fully explore the syntactic aspect of Chinese. As Chinese has very different word orders, a reordering approach, which can further explore the hierarchical structure of Chinese and utilize this information for reordering, may further improve the translation quality.

The hierarchical phrase-based translation model [13] is especially suitable for translation into Chinese, and delivers very good translation results. It extracts hierarchical rules by using information of the syntactic structure. Phrases from different hierarchies, or so-called phrases of phrases, are reordered during the decoding.

The idea of phrases on different hierarchies has inspired us to create this preordering method based on multiple levels of the syntax tree. Besides, we also hope to detach the reordering from decoding phase and do it separately in a pre-process before decoding, in order to reduce the time for translation. This kind of preordering approaches use linguistic information to modify the word orders.

Oracle reordering has also shown values for evaluating the potential of preordering. [14] introduced the permutation distance metrics which can be used to measure reordering quality. And [15] described how we can construct permutations from the word alignment as oracle reordering.

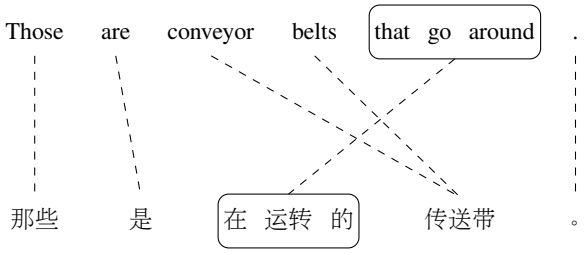


Figure 1: Position change of a relative clause

In this paper, we compare our results to the aforementioned rule-based reordering methods and the oracle reordering to get a better overview.

3. Motivation

The word order between English and Chinese differs very much. For one, the words in Chinese have generally different origins as those in English, which leads to very different vocabulary and word construction. Sometimes it is very hard to find corresponding words in the other language. For example, some prepositions in Chinese have very different usage than the corresponding prepositions in English. Also the continuous writing of Chinese without spaces makes this problem more severe, since word boundaries are not always so clear in Chinese. The text needs to be segmented first before translation. A word segmentation process is used to separate the words, but the results may not always be ideal.

For the other, both languages have sometimes very different sentence structures. Thus, a word-for-word translation between English and Chinese is often unnatural or difficult to understand. Each of them has some sentence patterns that do not exist or rarely used in the other. In Chinese, a modifier is often put before the part that it modifies. While in English, it is very common that the modifier is put after the part that it modifies. Besides, English sentences with a lot of long clauses may be more suitable to translate into several Chinese sentences, because in Chinese people do not tend to use long clauses in general.

Some typical problems of word orders between English and Chinese that we have found are as follows:

- *Pre-modifier instead of post-modifier*
  In Chinese people tend to use pre-modifier rather than post-modifier. This involves the position change of adverbials, relative clauses and preposition phrases during translation. Figure 1 shows an example of how the position of a relative clause changes

- *Construction of questions*
  The two languages have very different ways to con-

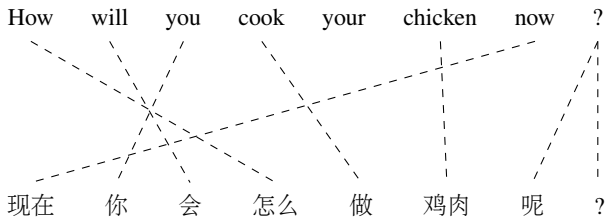


Figure 2: Word reordering of a question

struct questions, which raises word order problems for translating questions. Figure 2 shows word reordering of a question.

- *Special sentence constructions*
  For example, *Bâ*-construction (把字句) in Chinese and sentence constructions such as *there-be* and inverted negative sentences in English do not have correspondence in the other language in general.

- *Long distance word position change*
  Word reordering between English and Chinese often involves word shift cross long distance. For example, following translation shows that a adverbial clause (underlined) is shifted cross long distance when being translated.

  I find this very much disturbing when we are talking about what is going on right and wrong with democracy these days.

  现在，每当我跟别人讨论我们的民主什么是对的，什么是错的我都为此觉得很无力。

Following the analysis above, we can see the word reordering between English and Chinese is very unstructural, because it involves word position changes between different word groups and syntactic hierarchies in the source language. In order to improve the reordering, we need methods that can handle more complicated, unstructural word order change. Inspired by the ideas of reordering on syntax tree and hierarchical phrases, we created the Multiple-Level-Tree(MLT) reordering, which reorders words based on multiple syntactic levels and can handle long distance word position change and complicated word position change very well.

4. Multi-Level-Tree Reordering

Our preordering method is based on automatically learned reordering rules. Reordering rules show how sentences should be reordered in source language before translation. In our system, the rules are generated by using the word alignment, and syntax tree, all of which are calculated based on the training data. After reordering rules are applied to the source sentences, word lattices are generated. A word lattice contains all the reordering possibilities of a source sentence and is further passed to the decoder for translation. The preordering system is illustrated as Figure 3.

4.1. Reordering on Multiple Syntactic Levels

Reordering patterns are based on multiple levels of the syntax tree. Figure 4 illustrates how the reordering patterns are detected from the syntax tree. In the example, the detection starts from the root node, go downwards three levels and use the nodes in these levels to detect the reordering pattern. These nodes that are used for detecting the reordering pattern are colored gray and have an italic font. The nodes with dark gray are the lowest ones among such nodes, which represent constituents that are actually reordered through the reordering rules. The leaf nodes in the syntax tree indicate words in the sentence, which has rectangle shape with angular corners.

According to the alignment information with the corresponding Chinese translation, the node labeled with *NP* is moved forward to the first place in the translation and the node labeled with *IN-of* is moved forward to the second place in translation. This reordering cannot be handled with one-level tree-based reordering, but in MLT, from the root node with a search depth of three, the following reordering pattern can be found:

```
NP(CD0 NP(NP(JJ1 NNS2)PP(IN3 NP4)))

-> NP IN CD JJ NNS
-> 4 3 0 1 2 (alternative with index)
```

The POS tags in bold corresponds to nodes with dark gray in Figure 4, which presents constituents that are actually reordered. The parentheses in the reordering pattern indicate the corresponding hierarchies in the syntax tree. The reordering can be alternatively represented with indices, which is the actual internal representation to avoid ambiguity caused by POS tags with the same name and we use this representation throughout this paper.

4.2. Rule Extraction

In order to find as much information for reordering as possible, the algorithm of rule extraction detects the reordering patterns from all nodes in the syntax tree and it goes downwards for any number of hierarchies, until it reaches the lowest hierarchy in the subtrees.

In the implementation, the program conducts a depth-first search (DFS) to traverse every node in a syntax tree. Every time when a node is reached, the program conducts another iterative deepening depth-first search (IDDFS) in its subtree with depth-limit from 1 to the subtree's depth. And the program detects if there are any patterns of word position changes at the same time, by using the alignment for comparison.

The detected word position changes are checked for their validity for reordering rules. A valid reordering pattern should both have actual word reordering and clearly distinguishable new order on the side of the target language, i.e. no collision of aligned ranges on the target side.

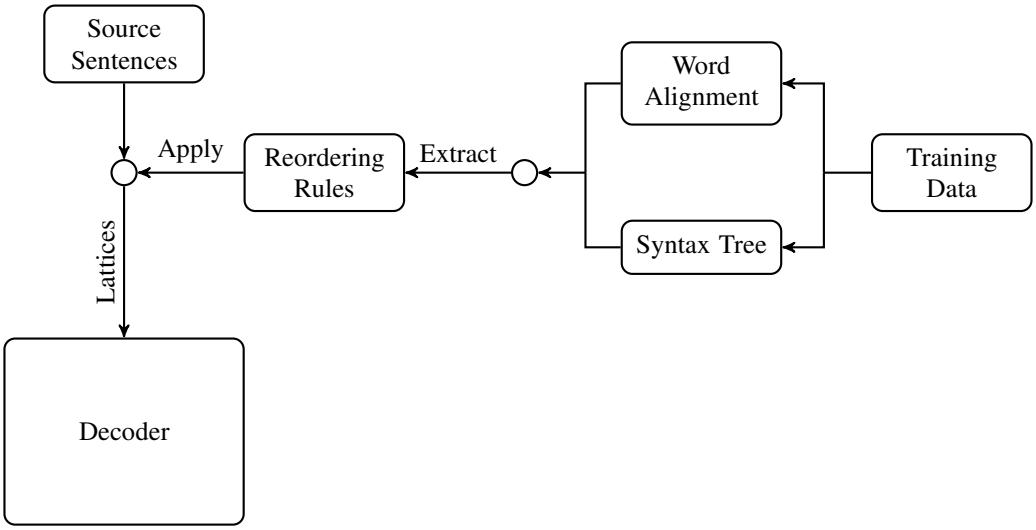


Figure 3: Illustration of preordering system

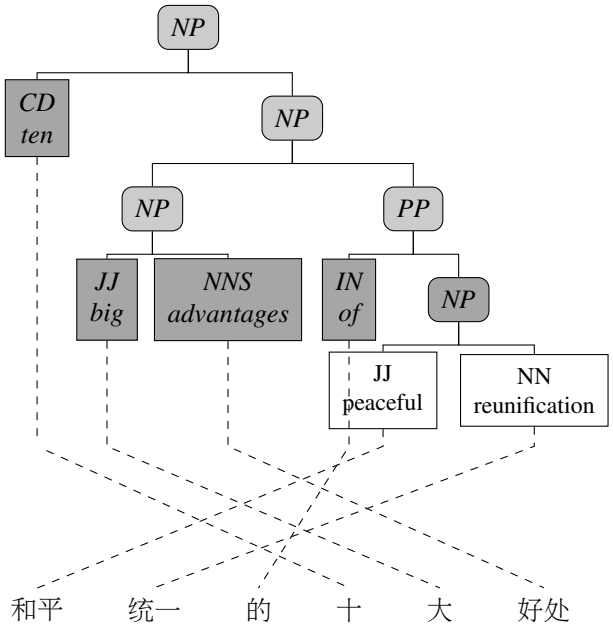


Figure 4: Detection of reordering pattern from multiple syntactic levels

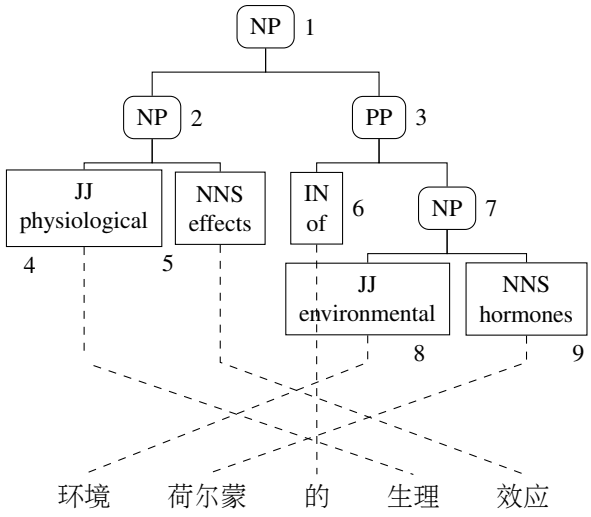


Figure 5: A phrase with its syntax tree and word alignment for rule extraction

Figure 5 shows a phrase to be translated, together with its syntax tree and word alignment of parallel text. In this example, we can find the following reordering patterns:

From node 1:

```
NP(NP PP)->1 0                                  [1 level]
NP(NP(JJ NNS)PP(IN NP))->3 2 0 1                [2 levels]
NP(NP(JJ NNS)PP(IN NP(JJ NNS)))->
3 4 2 0 1                                       [3 levels]
```

From node 3:

```
PP(IN NP)->1 0                                  [1 level]
PP(IN NP(JJ NNS))->1 2 0                        [2 levels]
```

The probability of the reordering patterns are calculated based on frequency of their occurrences in the training corpus. There are the left part and the right part of the reordering patterns separated by the arrow. The left part indicates the syntactic tags that should be reordered and the right part indicates how the new order should be like. The probability of the pattern is calculated by how often the left part is reordered into the right part among all its appearances in the training corpus. In addition, reordering patterns that appear less than a threshold are ignored to be used as reordering rules, in order to prevent too concrete rules without generalization capability and overfitting.

4.3. Rule Application

The syntax tree is traversed by DFS as the same in rule extraction. But from the root of each subtree, it has scanned with depth limit from its maximal levels, i.e. its depth, to 1. As it turns out that any rule can be applied for a subtree at some level, a new path for this reordering will be added to the word lattice for decoding. As long as rules can be applied on a subtree for a certain depth, the search for rule application on this subtree stops, and the search on the next subtree continues.

The reason for this is to prevent duplicate reorderings due to application of nested rules, which have overlapped effect with each other. These rules are normally patterns that are generated on the same subtree, but with different number of levels, which has different generalization effect on the same range of words in the text. For example, the following patterns can be detected from the syntax tree in Figure 5:

```
PP(IN NP) -> 1 0
PP(IN NP(JJ NNS)) -> 1 2 0
```

Both patterns are detected from the same node, but the second pattern is detected by retrieving the nodes one level deeper and it is more concrete. So the first pattern can be seen as a generalization of the second pattern. Whenever a rule of the second pattern can be applied, a rule of the first pattern can be applied too. Because subtrees are checked from the highest number of levels in rule application, the more concrete rule is applied first. Since the more concrete rule fits the detected pattern better and contains more details of reordering, so it may be more suitable for rule application. In this example, the second rule is applied rather than the first rule.

Word reorderings are added to the word lattice as paths, which is further past to the decoder for translation. Paths with very low probability are removed, in order to save space for storing the lattice and reduce decoding time later, without compromising too much translation quality.

4.4. Rule Combination

In order to further explore the probability of improvement, the MLT reordering rules can be combined with other types of reordering rules to achieve further improvements. This is done by training the different types of rules separately and applying them on the monotone path of the sentence independently. All the generated different paths are compressed in the word lattice.

5. Experiments

We have conducted experiments on both English-to-Chinese and Chinese-to-English translation directions to get a better overview of the MLT reordering's effect. In the experiments, the MLT reordering rules are also combined with other reordering rules that are introduced before, in order to show the improvement achieved by our approach.

	BLEU (%)	Imprv.	TER (%)
Baseline	12.07		72.15
+Short Rules	12.50	0.43	71.41
+Long Rules	12.99	0.92	70.71
+Tree Rules	13.38	1.31	68.27
+MLT Rules	**13.81**	**1.74**	**68.20**
Oracle Reordering	18.58	6.51	62.13
Long Rules	12.31	0.24	71.81
Tree Rules	13.30	1.23	70.42
MLT Rules	**13.68**	**1.61**	**70.25**

Table 2: Result overview of the English-to-Chinese system

In order to evaluate the potential of word reordering, we also used oracle reordering in the experiments. Oracle reordering is considered to be an optimally reordered sentence as input to the SMT system and do not allow additional reordering during decoding [12]. So we can use it as input of the SMT system and the scores are the optimal results that can be achieved by word reordering, from which we can evaluate the potential of reordering methods.

5.1. English-to-Chinese System

We performed experiments with and without different reordering methods covering the English-to-Chinese translation direction. The reordering methods included our MLT reordering approach and the other reordering approaches with short rules, long rules and tree rules. The system was trained on news text from the LDC corpus and subtitles from TED talks. The development data and test data were both news text from the LDC corpus. The system was a phrase-based SMT system, which used a 6-gram language model with Knersey-Ney smoothing. Besides the preordering, no lexical reordering or other reordering method in decoding phase was used. The text was translated through a monotone decoder.

The reordering rules were extracted by using the word alignments, POS tags and syntax trees from the training data. One reference of the test data was used for evaluating the results. The threshold for rule extraction is set as 5 times and reordering paths with probability less than 0.1 are not added to word lattices. The decoder was a monotone decoder. Table 1 shows the size of data used in the system.

Table 2 shows the BLEU scores, absolute improvements of BLEU scores and TER scores for configurations with different reordering methods. The table consists of 2 sections. the first row of the top section shows results of the baseline, which used no preordering at all. In the following rows of the top section, different types of reordering rules are combined gradually, with each type per row. For example, the row with *+MLT Rules* presents the configuration with all the rule types including MLT rules and all the other rules in the rows above. All improvements are absolute improvements of BLEU scores in comparison to the baseline. Each row

Data Set		Sentence Count	Word Count		Size (Byte)	
			English	Chinese	English	Chinese
Training Data	LDC	303K	10.96M	8.56M	60.88M	47.27M
	TED	151K	2.58M	2.86M	14.24M	15.63K
Development Data		919	30K	25K	164K	142K
Test Data		1663	47K	38K	263K	220K

Table 1: Corpus statistics in the English-to-Chinese system

	BLEU (%)	Imprv.	TER (%)
Baseline	21.80		62.09
+Short Rules	22.90	1.10	61.64
+Long Rules	23.13	1.33	61.43
+Tree Rules	23.84	2.04	60.95
+MLT Rules	**24.14**	**2.34**	**60.79**
Oracle Reordering	26.80	5.00	56.97
Long Rules	22.10	0.30	62.21
Tree Rules	23.35	1.55	61.52
MLT Rules	**23.96**	**2.16**	**60.83**

Table 4: Result overview of the Chinese-to-English systems

with a certain reordering type presents all the different variations of this type and the best score under these configurations is shown. For example, long rules include the left rules and right rules, and the tree rules include the partial rules and recursive application. The baseline used a monotone decoder and no preordering. The row with *oracle reordering* shows the results from the configuration that used the oracle reordering as input. The results of oracle reordering can be used for analyzing the potential of source sentence reordering. In the lower section of the table, different rule types are not combined and the effect of each rule type is shown.

5.2. Chinese-to-English System

The experiments for Chinese-to-English systems have a similar setup as described in the last section. The parallel data used in the English-to-Chinese system was also used in this experiment by switching the source language and the target language. We only used the LDC data set for training, and no TED data were used in this system. The test data had three English references for evaluating the results instead of one as in the previous system. The data used are summarized in Table 3.

Table 4 shows the results for configurations with different reordering methods for the Chinese-to-English translation. The table can be interpreted in the same manner as Table 2 in the previous section.

5.3. Evaluation

The results shows increasing scores when we used reordering methods from short rules, long rules, tree rules to MLT rules. And better BLEU scores were achieved when we combined the different reordering rules. The MLT rules achieved better BLEU scores and TER scores in both translation directions, not only when it was used alone, but also it was added to the other reordering rules. As the MLT reordering rules were used alone, it showed an improvement of 1.61 in BLEU score in the English-to-Chinese translation direction and an improvement of 2.16 in BLEU score in the Chinese-to-English translation direction, in comparison with the baseline, which used no reordering at all. As the MLT reordering rules were combined with the other existing reordering rules, a further improvement of 0.43 in BLEU score (from 13.38 to 13.81) was shown in the English-to-Chinese translation direction, as well as a further improvement of 0.3 in BLEU score (from 23.84 to 24.14) in the Chinese-to-English translation direction.

We have also found improvements in the sentence structure. Table 5 and Table 6 show some translation examples in both translation directions. Sections are separated by double lines in the table. Each section of this table shows one translation example with the source sentence (*source*), translation without using MLT reordering (*no MLT*), translation with MLT reordering (*MLT*) and the reference (*reference*). *Glossary* for the source sentences or references in Chinese is also added as word for word translation. Each word or group of hyphenated words in the glossary corresponds a Chinese character or a group of Chinese characters that are not separated with space. A placeholder □ is used to replace words that are difficult to translate, which play grammatical roles in general. The translation without using MLT reordering comes from the configuration with highest BLEU score that did not use MLT reordering. And the translation with MLT reordering comes from the configuration with highest BLEU score that used MLT reordering. From the examples, we can clearly see the improvements in sentence structure.

From these experiments we can draw the conclusion that our reordering method obviously improves the sentence structure and translation quality in both English-to-Chinese and Chinese-to-English translation directions, no matter when we apply it alone or when we combine it with short rules, long rules and tree rules.

6. Conclusions

We have presented a new preordering approach for translation between English and Chinese. The algorithm detects and applies reordering rules by using information of multi-

Data Set	Sentence Count	Word Count		Size (Byte)	
		Chinese	English	Chinese	English
Training Data	303K	8.56M	10.96M	47.27M	60.88M
Development Data	919	25K	30K	142K	164K
Test Data	1663	38K	47K	220K	263K

Table 3: Corpus statistics in the Chinese-to-English system

Source	陈至立 说 , 古巴 是 拉美 和 加勒比 地区 有 重要 影响 的 国家 。
Glossary	chen-zhili said , cuba is latin-american and caribbean region has great influence □ country .
No MLT	chen zhili said : cuba is the latin america and the caribbean region has an important influence on the state .
MLT	chen zhili said : cuba is a country of important influence latin america and the caribbean region .
Reference	chen zhili said that cuba is a country of great influence in the latin american and caribbean region .
Source	近年 来 , 两 国 教育 交流 日益 密切 , 人员 来往 频繁 。
Glossary	recent-years in, two countries educational exchange increasingly close , personnel visits frequent .
no MLT	in recent years , the two countries education have been increasingly close exchanges and personnel contacts have been frequent .
MLT	in recent years , the educational exchanges between the two countries have become increasingly frequent , and have had frequent contacts .
Reference	in recent years , the educational exchange between the two countries has become increasingly close with frequent personnel visits .

Table 5: Examples of translations from Chinese to English

Source	hu jintao also extended deep condolences on the death of the chinese victims and expressed sincere sympathy to the bereaved families .
No MLT	胡锦涛 还 表示 深切 哀悼 的 受害者 家属 的 死亡 , 向 迁难者 家属 表示 诚挚 的 慰问 。
MLT	胡锦涛 还 对 中国 迁难者 表示 哀悼 , 向 迁难者 家属 表示 诚挚 的 慰问 。
Reference	胡锦涛 还 对 中方 不幸 遇难 人员 表示 深切 的 哀悼 , 并 向 遇难者 的 亲属 致以 诚挚 的 慰问 。
Glossary	hu-jintao also on chinese unfortunately killed people extended deep □ condolences , and to bereaved 's families expressed sincere □ sympathy .
Source	satisfying personal interests and expanding knowledge are also major reasons why hourly work appeals to people .
No MLT	满足 个人 利益 和 扩大 知识 也 是 主要 原因 小时 工作 吸引 人 。
MLT	满足 个人 利益 和 扩大 知识 也 是 为什么 学生 工作 吸引 人 的 主要 原因 。
Reference	满足 个人 兴趣 , 扩大 自己 的 知识面 也 是 兼职 小时 工 受 青睐 的 一 个 重要 原因 。
Glossary	satisfying personal interests , expanding own □ knowledge also are part-time hourly work is favored □ one □ major reason .
Source	the dalai lama will go to visit washington this month .
No MLT	达赖 喇嘛 将 访问 华盛顿 的 这 一 个 月 。
MLT	达赖 喇嘛 将 本 月 访问 华盛顿 。
Reference	达赖 喇嘛 将 在 本 月 前往 华盛顿 访问 。
Glossary	dalai lama will in this month go-to washington visit .

Table 6: Examples of translations from English to Chinese

ple syntactic levels in the syntax tree and word alignment to reorder the source sentences. Reordering patterns are detected by checking if the nested tag sequences in subtrees with any number of search levels have clearly new orders in the aligned text in the target language.

We have conducted experiments in both translation directions between English and Chinese with different SMT configurations. From the results we can see the BLEU scores were improved no matter when we applied the SMT reordering method to the baseline directly or when we combined it with the other reordering methods, i.e. short rules, long rules and tree rules based reordering methods.

Besides the improvement in BLEU scores, our preordering approach also showed improvement in the sentence structure of the translation. Considering sentences that have complicated structure only make up a small part of the data, even a small improvement in the result can mean a big improvement in translating these complicated sentences.

By taking a close look at the gap between the scores of oracle reordering and the best scores achieved by MLT reordering, we can also see, there is still potential for improvements of translation between English and Chinese through better reordering methods.

7. Acknowledgments

The authors gratefully acknowledge the support by an interACT student exchange scholarship. The research leading to these results has received funding from the European Union 7th Framework Programme (FP7/2007-2013) under grant agreement No. 287658.

8. References

[1] Rottmann, K. and Vogel, S., "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *TMI*, (Skövde, Sweden), 2007.

[2] Niehues, J. and Kolss, M., "A POS-Based Model for Long-Range Reorderings in SMT," in *Proceedings of the Fourth Workshop on Statistical Machine Translation*, (Athens, Greece), pp. 206–214, Association for Computational Linguistics, 2009.

[3] Herrmann, T., Weiner, J., Niehues, J., and Waibel, A., "Analyzing the Potential of Source Sentence Reordering in Statistical Machine Translation," in *IWSLT*, (Heidelberg, Germany), 2013.

[4] Tillmann, C., "A Unigram Orientation Model for Statistical Machine Translation," in *Proceedings of HLT-NAACL 2004: Short Papers*, pp. 101–104, Association for Computational Linguistics, 2004.

[5] Koehn, P., Axelrod, A., Birch, A., Callison-Burch, C., Osborne, M., Talbot, D., and White, M., "Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation," in *IWSLT*, pp. 68–75, 2005.

[6] Collins, M., Koehn, P., and Kučerová, I., "Clause Restructuring for Statistical Machine Translation," in *Proceedings of the 43rd annual meeting on association for computational linguistics*, pp. 531–540, Association for Computational Linguistics, 2005.

[7] Popovic, M. and Ney, H., "POS-Based Word Reorderings for Statistical Machine Translation," in *International Conference on Language Resources and Evaluation*, pp. 1278–1283, 2006.

[8] Habash, N., "Syntactic Preprocessing for Statistical Machine Translation," *MT Summit XI*, pp. 215–222, 2007.

[9] Wang, C., Collins, M., and Koehn, P., "Chinese Syntactic Reordering for Statistical Machine Translation," in *EMNLP-CoNLL*, pp. 737–745, Citeseer, 2007.

[10] Zhang, Y., Zens, R., and Ney, H., "Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation," in *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, pp. 1–8, Association for Computational Linguistics, 2007.

[11] Crego, J. M. and Habash, N., "Using Shallow Syntax Information to Improve Word Alignment and Reordering for SMT," in *Proceedings of the Third Workshop on Statistical Machine Translation*, pp. 53–61, Association for Computational Linguistics, 2008.

[12] Herrmann, T., Niehues, J., and Waibel, A., "Combining Word Reordering Methods on Different Linguistic Abstraction Levels for Statistical Machine Translation," in *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, (Atlanta, Georgia), pp. 39–47, Association for Computational Linguistics, June 2013.

[13] Chiang, D., "Hierarchical Phrase-Based Translation," *computational linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[14] Birch, A., Osborne, M., and Blunsom, P., "Metrics for MT Evaluation: Evaluating Reordering," *Machine Translation*, vol. 24, Mar. 2010.

[15] Birch, A., "Reordering Metrics for Statistical Machine Translation," 2011. PhD thesis.

Index

iwslt2014.org