International Workshop on Spoken Language Translation

December 3–4, 2015

Proceedings

Da Nang, Vietnam

iwslt.org

Proceedings of the

International Workshop on Spoken Language Translation

December 3-4, 2015

Da Nang, Vietnam

Edited by
Marcello Federico
Sebastian Stüker
Jan Niehues

Contents

Foreword

The International Workshop on Spoken Language Translation (IWSLT) is an annually scientific workshop, associated with an open evaluation campaign on Spoken Language Translation, where both scientific papers and system descriptions are presented. Since 2004, the annual workshop has been held in Kyoto, Pittsburgh, Kyoto, Trento, Honolulu, Tokyo, Paris, San Francisco, Hong Kong, Heidelberg and Lake Tahoe and this year, the 12th International Workshop on Spoken Language Translation takes place in Da Nang, Vietnam on Dec. 03 and 04, 2015.

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
- Spoken language summarization
- 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 members of the program committee and papers were selected based on their technical merit and relevance to the conference. 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. 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 to more practical issues related to the integration of speech and translation technologies.

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 is organized in Vietnam and one of the purposes is to promote Spoken Language Translation research activities in Asian countries. More languages are added in this scientific program of IWSLT including Vietnamese-to-English, Filipino-to-English bidirectional statistical Machine Translation Systems.

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 his system, the utilized resources. A survey of the evaluation campaigns is presented by the organizers.

Apart from the technical content of the workshop, beautiful beaches, fresh sea food, authentic local cuisines, world class service and the artistic ocean front villas of Da Nang will welcome all participants to the third largest city in Vietnam.

Welcome to Da Nang!
Luong Chi Mai

Organizers

Chairs

Chi Mai Luong (IOIT, Vietnam): Workshop
Marcello Federico (FBK, Italy): Evaluation Committee
Sebastian Stüker (KIT, Germany): Evaluation Committee
Jan Niehues (KIT,Germany): Program Committee
Alex Waibel (CMU, USA/KIT, Germany): Steering Committee

Local and Financial Chair

Chi Mai Luong (IOIT, Vietnam)

Evaluation Technical Committee

Sebastian Stüker (KIT, Germany): ASR Track
Mauro Cettolo (FBK, Italy): MT Track
Jan Niehues (KIT, Germany): SLT Track
Luisa Bentivogli (FBK, Italy): Human Evaluation
Roldano Cattoni (FBK, Italy): Evaluation Server

Steering Committee

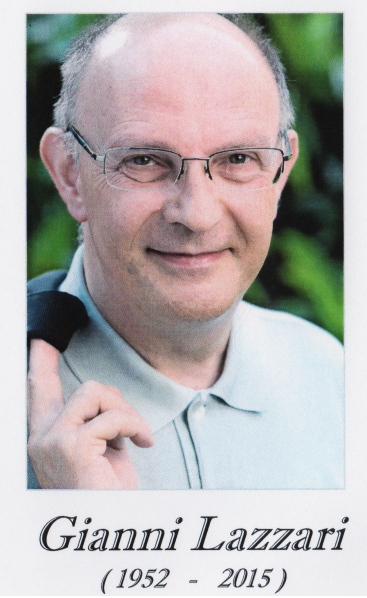
Alex Waibel (CMU, USA/KIT, Germany): Co-Chair
Marcello Federico, (FBK, Italy): Co-Chair
Hisashi Kawai (NICT, Japan)
Joseph Mariani (IMMI, France)
Satoshi Nakamura (NAIST, Japan)
Sebastian Stüker (KIT, Germany)
Hermann Ney (RWTH, Germany)
Francois Yvon (CNRS-LIMSI, France)
Chi Mai Luong (IOIT, Vietnam)

Program Committee

Gilles Adda (LIMSI/CNRS, FR)
Alexandre Allauzen (LIMSI/CNRS, FR)
Loïc Barrault (LIUM,FR)
Laurent Besacier (Laboratoire d'Informatique de Grenoble, FR)
Francisco Casacuberta (Universitat Politècnica de València, ES)

Mauro Cettolo (FBK, IT)
Boxing Chen (NRC-CNRC, CA)
Eunah Cho (KIT,DE)
Markus Freitag (RWTH, DE)
Francisco Javier Guzman (QCRI, QA)
Thanh-Le Ha (KIT, DE)
Eva Hasler (University of Cambridge, UK)
Michael Heck (Nara Institute of Science and Technology, JP)
Teresa Herrmann (KIT, DE)
Matthias Huck (University of Edinburgh, UK)
Kevin Kilgour (KIT, DE)
Yves Lepage (Waseda University, JP)
Mohammed Mediani (KIT, DE)
Markus Mueller (KIT, DE)
Graham Neubig (Nara Institute of Science and Technology, JP)
Jan Niehues (KIT, DE)
Markus Nußbaum-Thom (RWTH, DE)
Nicolas Pecheux (LIMSI/CNRS, FR)
Stephan Peitz (RWTH, DE)
Jan-Thorsten Peter (RWTH ,DE)
Nicholas Ruiz (FBK, IT)
Rico Sennrich (University of Edinburgh, UK)
Christophe Servan (Laboratoire d'Informatique de Grenoble,FR)
Matthias Sperber (KIT, DE)
Sebastian Stüker (KIT, DE)
Isabel Trancoso (INESC ID Lisboa / IST, PT)
Joachim Van Den Bogaert (Cross Language, BE)
Marion Weller (Universität Stuttgart, DE)
Guillaume Wisniewski (LIMSI/CNRS, FR)
Dekai Wu (HKUST, CN)
François Yvon (LIMSI/CNRS, FR)
Yuqi Zhang (KIT, DE)

Dedication



The organisers wish to dedicate this workshop to the memory of **Gianni Lazzari**, who left us on the 26th of November. Gianni was among the initial promoters of IWSLT, and actively participated to its organisation in the first years. From 1985 to 2007, Gianni was research division director at ITC-IRST in Trento (Italy) and oversaw, over the years, IRST's research efforts in speech recognition, speech translation, signal processing, computer vision, and predictive models. In these fields he collaborated in and coordinated a number of international projects. In few years he was able to position his organisation as a point of excellence at national level. With his strong inclination for innovation, he also fostered the creation of several successful spin-off companies. His natural cordiality allowed him to make many friends around the world and in his beloved Trentino. In 2007, he took on a new challenge as CEO of Habitech, the new provincial district for energy and environment, which he developed as an exemplar bridge between innovation and industry. In 2013, he was diagnosed with blood cancer, but after long treatments he was able to go back to work. Until few weeks ago, when the disease hit him definitely. He leaves behind his beloved wife Chiara and daughter Cecilia. We all miss a good friend, who greatly contributed to our research field and, more importantly, enriched our personal lives.

Acknowledgments

IWSLT 2015 is proud to present its gold sponsor

Participants

	ASR		SLT			
	en	de	en − de	de − en	en − fr	en − zh
HLT-I2R (VIETNAM)	✔					
IOIT (VIETNAM)	✔					
KIT (GERMANY)	✔	✔	✔	✔		
LIUM (FRANCE)		✔			✔	
MITLL-AFRL (USA)	✔					✔
MLLP (SPAIN)	✔	✔				
NAIST (JAPAN)	✔					
RWTH (GERMANY)				✔		

Groups participating to the ASR and SLT evaluation tasks

	en − de	en − fr	en − vi	en − zh	en − cz
HDU (GERMANY)	⇒				
JAIST (JAPAN)			⇐,⇒		
KIT (GERMANY)	⇐,⇒		⇒		
MITLL-AFRL (USA)				⇐,⇒	
PJAIT (POLAND)	⇐,⇒	⇐,⇒	⇐,⇒		⇐,⇒
UEDIN (UK)	⇒			⇒	
UMD (USA)		⇐	⇐		
UNETI (VIETNAM)			⇐,⇒		
SU (USA)	⇒		⇒		
RWTH (GERMANY)	⇐				
TUT (JAPAN)			⇐		

Groups participating to the MT evaluation tasks

Program

Thursday, December 3rd, 2015

08:30-09:15	**WORKSHOP REGISTRATION**
09:15-09:30	**Welcome Remarks** *Chi Mai Luong (General Chair) & Alex Waibel (Steering Committee Chair)* IOIT, Vietnam & KIT-CMU, Germany, USA
09:30-10:30	**Report on the 12th IWSLT Evaluation Campaign, IWSLT 2015** *Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli and Marcello Federico* FBK, Italy
Coffee Break (10:30-11:00)	
11:00-12:30	**EVALUATION CAMPAIGN**
11:00-11:30	**ASR:The I2R ASR System for IWSLT 2015** *Huy Dat Tran, Jonathan Dennis and Wen Zheng Ng* HLT-I2R, Singapore
11:30-12:00	**SMT: Stanford Neural Machine Translation Systems for Spoken Language Domains** *Minh-Thang Luong and Christopher D. Manning* Stanford, USA
12:00-12:30	**SLT: The LIUM ASR and SLT Systems for IWSLT 2015** *Mercedes Garcia Martinez, Loïc Barrault, Anthony Rousseau, Paul Deléglise and Yannick Estève* LIUM, France
Lunch (12:30-14:00)	
14:00-16:00	**ORAL SESSION I**
14:00-14:30	**Multifeature Modular Deep Neural Network Acoustic Models** *Kevin Kilgour and Alex Waibel* KIT, Germany
14:30-15:00	**Class-Based N-gram Language Difference Models for Data Selection** *Amittai Axelrod, Yogarshi Vyas, Marianna Martindale and Marine Carpuat* JHU/UMD, USA
15:00-15:30	**Parser Self-Training for Syntax-Based Machine Translation** *Makoto Morishita, Koichi Akabe, Yuto Hatakoshi, Graham Neubig, Koichiro Yoshino and Satoshi Nakamura* NAIST, Japa
Coffee Break (15:30-16:00)	

16:90-17:30	**POSTER SESSION I**
	Unsupervised comparable corpora preparation and exploration for bilingual translation equivalents *Krzysztof Wolk and Krzysztof Marasek* PJIIT, Poland
	Evaluation and Revision of a Speech Translation System for Healthcare *Mark Seligman and Mike Dillinger* Spoken Translation, Inc., USA
	Source Discriminative Word Lexicon for Translation Disambiguation *Teresa Herrmann, Jan Niehues and Alex Waibel.* KIT, Germany
	Using Language Adaptive Deep Neural Networks for Improved Multilingual Speech Recognition *Markus Mueller and Alex Waibel* KIT,Germany
	Morphology-Aware Alignments for Translation to and from a Synthetic Language *Franck Burlot and François Yvon* LIMSI-CNRS, France
	The IOIT English ASR system for IWSLT 2015 *Van Huy Nguyen, Quoc Bao Nguyen, Tat Thang Vu and Chi Mai Luong* IOIT, Vietnam
	The I2R ASR System for IWSLT 2015 *Huy Dat Tran, Jonathan Dennis and Wen Zheng Ng* HLT-I2R, Singapore
	The NAIST English Speech Recognition System for IWSLT 2015 *Michael Heck, Quoc Truong Do, Sakriani Sakti, Graham Neubig and Satoshi Nakamura* NAIST, Japan
	The RWTH Aachen Machine Translation System for IWSLT 2015 *Jan-Thorsten Peter, Farzad Toutounchi, Stephan Peitz, Parnia Bahar, Andreas Guta and Hermann Ney* RWTH, Germany
	The MITLL-AFRL IWSLT 2015 MT System *Michaeel Kazi, Brian Thompson, Elizabeth Salesky, Timothy Anderson, Grant Erdmann, Eric Hansen, Brian Ore, Katherine Young, Jeremy Gwinnup, Michael Hutt and Christina May* MITLL-AFRL, USA
	The MLLP ASR Systems for IWSLT 2015 *Miguel Ángel Del Agua Teba, Adrià Agustí Martínez Villaronga, Santiago Piqueras Gozalbes, Adrià Giménez Pastor, Jose Alberto Sanchís Navarro, Jorge Civera Saiz and Alfons Juan Císcar* MLLP, Spain
	The LIUM ASR and SLT Systems for IWSLT 2015 *Mercedes Garcia Martinez, Loïc Barrault, Anthony Rousseau, Paul Deléglise and Yannick Estève* LIUM, France

	The KIT Translation Systems for IWSLT 2015 *Thanh-Le Ha, Jan Niehues, Eunah Cho, Mohammed Mediani and Alex Waibel* KIT, Germany
	The 2015 KIT IWSLT Speech-to-Text Systems for English and German *Markus Mueller, Thai Son Nguyen, Matthias Sperber, Kevin Kilgour, Sebastian Stüker and Alex Waibel* KIT, Germany
18:30-	**SOCIAL EVENT DINER**

Friday, December 4th, 2014

09:00-10:30	**ORAL SESSION II**
9:00-10:00	**Invited talk: Improving SMT by Model Filtering and Phrase Embedding** *Chengqing ZONG* NLPR, China
10:00-10:30	**Open Discussion: The future of IWSLT evaluation** *Chair: Marcello Federico and Sebastian Stüker* FBK, Italy and KIT, Germany
Coffee Break (10:30-11:00)	
11:00-12:30	**ORAL SESSION III**
11:00-11:30	**Inducing Bilingual Lexicons from Small Quantities of Sentence-Aligned Phonemic Transcriptions** *Oliver Adams, Graham Neubig, Trevor Cohn and Steven Bird* UNIMELB, Australia; NAIST, Japan
11:30-12:00	**Learning Segmentations that Balance Latency versus Quality in Spoken Language Translation** *Hassan S. Shavarani, Maryam Siahbani, Ramtin Mehdizadeh Seraj and Anoop Sarkar* SFU, Canada
12:00-12:30	**Punctuation Insertion for Real-time Spoken Language Translation** *Eunah Cho, Jan Niehues and Alex Waibel* KIT, Germany
Lunch (12:30-14:00)	
14:00-16:00	**ORAL SESSION: IV**
14:00-15:00	**Panel: New trends in Spoken Language translation**
15:00-15:30	**An Open Source Toolkit for Word-level Confidence Estimation in Machine Translation** *Christophe Servan, Le Ngoc Tien, Luong Ngoc Quang, Lecouteux Benjamin and Besacier Laurent* IMAG, France; IDIAP, Switzerland
15:30-16:00	**Phrase-level Quality Estimation for Machine Translation** *Varvara Logacheva and Lucia Specia* USFD, UK
Coffee Break (16:00-16:30)	
16:30-18:00	**POSTER SESSION II**
	Development of a Filipino-to-English Bidirectional Statistical Machine Translation System that dynamically updates via user feedback *Joel Ilao, Jasmine Ang, Marc Randell Chan, Paolo Genato and Joyce Uy* DLSU, Philippines
	Risk-aware Distribution of SMT Outputs for Translation of Documents Targeting Many Anonymous Readers *Yo Ehara, Masao Utiyama and Eiichiro Sumita* TMU/NICT, Japan

	Applying Cross-Entropy Difference for Selecting Parallel Training Data from Publicly Available Sources for Conversational Machine Translation *William Lewis, Christian Federmann and Ying Xin* Microsoft Research, USA
	Improving Continuous Space Language Models using Auxiliary Features *Walid Aransa, Holger Schwenk and Loic Barrault* LIUM, France
	Improving Translation of Emphasis with Pause Prediction in Speech-to-speech Translation Systems *Quoc Truong Do, Sakriani Sakti, Graham Neubig, Tomoki Toda and Satoshi Nakamura* ICTS/NAIST, Japan
	The English-Vietnamese Machine Translation System for IWSLT 2015 *Viet Tran Hong, Huyen Vu Thuong, Vinh Nguyen Van and Trung Le Tien* Hanoi, Vietnam
	The JAIST-UET-MITI Machine Translation Systems for IWSLT 2015 *Hai-Long Trieu, Thanh-Quyen Dang, Phuong-Thai Nguyen and Le-Minh Nguyen* JAIST,Japan /UET,Vietnam
	PJAIT Systems for the IWSLT 2015 Evaluation Campaign Enhanced by Comparable Corpora *Krzysztof Wolk and Krzysztof Marasek* PJAIT, Poland
	Improvement of Word Alignment Models for Vietnamese-to-English Translation *Takahiro Nomura, Hajime Tsukada and Tomoyoshi Akiba* TUT, Japan
	The Edinburgh Machine Translation Systems for IWSLT 2015 *Matthias Huck and Alexandra Birch* UEDIN, UK
	The Heidelberg University English-German translation system for IWSLT 2015 *Laura Jehl, Patrick Simianer, Julian Hitschler and Stefan Riezler* HDU, Germany
	The UMD Machine Translation Systems at IWSLT 2015 *Amittai Axelrod and Marine Carpuat* UMD, USA
	Stanford Neural Machine Translation Systems for Spoken Language Domains *Minh-Thang Luong and Christopher D. Manning* Stanford, USA
18:00-	**CLOSING REMARKS + ANNOUNCEMENTS**

Keynotes

Improving SMT by Model Filtering and Phrase Embedding

Chengqing Zong, National Laboratory of Pattern Recognition

Abstract

In phrase-based and hierarchical phrase-based statistical machine translation systems, translation performance is heavily dependent on the size and quality of the translation table. To meet the requirements of enabling real-time responses, some research has focused upon filtering (pruning) of the translation table. However, most existing filtering methods have been based on application of one or two constraints that act as hard rules, such as disallowing phrase-pairs with low translation probabilities. These approaches sometimes result in rigid constraints because they consider only a single factor instead of composite factors. In view of the above considerations, I will introduce a machine learning-based framework that integrates multiple features when pruning translation models. In addition, to improve the performance of phrase-based translation models, I will propose Bilingually-constrained Recursive Auto-encoders (BRAEs) for learning semantic phrase embeddings (compact vector representations for phrases), capable of distinguishing phrases with different semantic meanings. This method has been evaluated on two end-to-end SMT tasks and shows remarkable effectiveness on both tasks.

Bio

Chengqing Zong received his Ph.D. from the Institute of Computing Technology of the Chinese Academy of Sciences, in March, 1998. From May, 1998 to April, 2000 he worked as a post-doctoral researcher at the National Laboratory of Pattern Recognition (NLPR) in the Institute of Automation of the Chinese Academy of Sciences. He joined the NLPR in April, 2000, and is now a professor. In 1999 and 2001 he visited Japan's Advanced Telecommunications Research Institute International (ATR) as a guest researcher. From October, 2004 to February, 2005 he visited CLIPS-IMAG in France. His research interests include machine translation, natural language processing, and sentiment classification. He has authored a book and published more than 100 papers. He is a member of the International Committee on Computational Linguistics (ICCL) and the chair of Special Interest Group on Chinese Language Processing (SIGHAN) of the Association for Computational Linguistics (ACL). He is an associate editor of ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) and an editorial board member of IEEE Intelligent Systems, Machine Translation, Journal of Computer Science and Technology. He also served the

ACL-IJCNLP 2015 conference as a programming committee co-chair, and has helped organize many other conferences, including IJCAI, COLING, EMNLP and WWW etc., as a programming committee member or in other leadership positions.

Evaluation Campaign

The IWSLT 2015 Evaluation Campaign

M. Cettolo[(1)] J. Niehues[(2)] S. Stüker[(2)] L. Bentivogli[(1)] R. Cattoni[(1)] M. Federico[(1)]

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

Abstract

The IWSLT 2015 Evaluation Campaign featured three tracks: automatic speech recognition (ASR), spoken language translation (SLT), and machine translation (MT). For ASR we offered two tasks, on English and German, while for SLT and MT a number of tasks were proposed, involving English, German, French, Chinese, Czech, Thai, and Vietnamese. All tracks involved the transcription or translation of TED talks, either made available by the official TED website or by other TEDx events. A notable change with respect to previous evaluations was the use of unsegmented speech in the SLT track in order to better fit a real application scenario. Thus, from one side participants were encouraged to develop advanced methods for sentence segmentation, from the other side organisers had to cope with the automatic evaluation of SLT outputs not matching the sentence-wise arrangement of the human references. A new evaluation server was also developed to allow participants to score their MT and SLT systems on selected dev and test sets. This year 16 teams participated in the evaluation, for a total of 63 primary submissions. All runs were evaluated with objective metrics, and submissions for two of the MT translation tracks were also evaluated with human post-editing.

1. Introduction

We present the results of the 2015 evaluation campaign organized by the International Workshop of Spoken Language Translation. The IWSLT evaluation has been running for twelve years and has been offering a variety of speech recognition, speech translation and text translation tasks [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

The 2015 IWSLT evaluation focused on the automatic transcription and translation of TED and TEDx talks, i.e. public speeches covering many different topics. The evaluation included three tracks:

- 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.

As a major difference from the previous editions, not only participants in the ASR track but also those participating in the SLT track had to cope with *unsegmented speech* instead of pre-segmented speech. Thus, both ASR and SLT systems had to face the more realistic working condition of transcribing and translating a speech signal corresponding to an entire talk rather than a sequence of isolated speech segments, as in the past editions.

This year, the ASR track was on two languages, namely English and German. The SLT track included German to English and English to Chinese, Czech, French, German, Thai, and Vietnamese; the MT track offered the same tasks as SLT but in both directions.

For all tasks, all permissible training data sets were specified and instructions for the submissions of test runs were given together with the detailed evaluation schedule. This year, parallel data made available to the participants included an updated version of the WIT^3 [12] corpus of TED talks, data from the WMT 2015 shared tasks, the MULTIUN corpus, and Wikipedia translations kindly made available by PJAIT[13].

The test sets used for this year's evaluation (tst2015) include new TED or TEDx talks not previously released. Furthermore, for the ASR and MT tasks offered both in 2014 and 2015, progresses were assessed by asking participants to run their systems also on the test sets of edition 2014 (tst2014), which were specifically released again to this purpose.

All runs submitted by participants were evaluated with automatic metrics. In particular, for the SLT and MT tracks, an evaluation server was set up so that participants could autonomously score their runs on different dev and test sets. For two MT tasks, English-German and Vietnamese-English, systems were also evaluated by calculating HTER values on post-edits created by professional translators.

This year, 16 groups participated in the evaluation (see Table 1) submitting a total of 63 primary runs: 18 to the ASR track (9 for tst2015 and 9 for tst2014), 5 to the SLT track, and 40 to the MT track (26 for tst2015 and 14 for tst2014).

In the following, we overview each of the offered tracks (Sections 2, 3, 4), whose detailed results are provided in Appendix A. We also report on human evaluation (Section 5 and Appendix B), and finally draw some conclusions.

Proceedings of the 12th International Workshop on Spoken Language Translation
Da Nang, Vietnam, December 3-4, 2015

Table 1: List of Participants

UNETI	University Of Economic And Technical Industries, Vietnam [14]
IOIT	Institute of Information Technology, Vietnam [15]
HLT-I2R	Institute for Infocomm Research, Singapore [16]
JAIST	Japan Advanced Inst. of Sc. and Technology; U. of Eng. and Technology; MITI [17]
PJAIT	Polish-Japanese Academy of Information Technology, Poland [13]
NAIST	Nara Institute of Science and Technology, Japan [18]
TUT	Toyohashi University of Technology, Japan [19]
RWTH	Rheinisch-Westfälische Technische Hochschule Aachen, Germany [20]
MITLL-AFRL	MIT Lincoln Laboratory and Air Force Research Laboratory, USA [21]
UEDIN	University of Edinburgh, United Kingdom [22]
MLLP	Machine Learning and Language Processing Research Group, Spain [23]
HDU	Dept. of Computational Linguistics, Heidelberg University, Germany [24]
LIUM	Laboratoire d'Informatique de l'Université du Maine, France [25]
UMD	University of Maryland, USA [26]
KIT	Karlsruhe Institute of Technology, Germany [27, 28]
SU	Stanford University, USA [29]

2. ASR Track

2.1. Definition

The goal of the *Automatic Speech Recognition* (ASR) track for IWSLT 2015 was to transcribe English TED and German TEDx talks. The speech in TED lectures is in general planned, well articulated, and recorded in high quality. Actually TED talks are often rehearsed rigorously for several days with experts advising on and designing the presentation. Thus, to a certain degree, they almost resemble a stage performance. 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 that is often used in order to make talks entertaining. For the TEDx talks the recording conditions are often more difficult than for the English TED talks, as recording is usually done with a lower budget with worse equipment and less trained personnel. While the TEDx talks aim to mimic the TED talks, they are not as well prepared and well rehearsed as the TED lectures, thus portraying a more difficult to recognize speaking style and more adverse recording conditions for ASR.

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 to the spoken language translation evaluation (SLT), see Section 3.

2.2. Evaluation

Participants had to submit the results of the recognition of the tst2015 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.

For German, the transcriptions of the talks were generated manually by trained transcribers at KIT, while the initial English transcripts were derived from the subtitles available via TED by performing a forced alignment of the subtitles to the audio file. Then, a fast manual check was performed by listening to the talk and simultaneously scanning the aligned transcripts. In this way major deviations of the subtitles from the audio were detected. The subtitles were then either manually corrected or the affected portions of the audio were excluded from scoring. The more subtle differences between the subtitles and the actual spoken words were left for detection during the adjudication phase.

In order to measure the progress of the systems over the years, participants also had to provide results on the test set from 2014, i.e. *tst2014*.

2.3. Submissions

For this year's evaluation we received primary submissions from seven sites. For English we received six primary runs on tst2015 and six on tst2014, while for German we received 3+3 primary submissions. For English we further received a total of five contrastive submissions from three sites.

2.4. Results

The detailed results of the primary submissions of the evaluation in terms of word error rate (WER) can be found in Appendix A. The word error rates of the submitted systems on tst2015 are in the range of 6.6%–13.8% for English and 17.6%–43.3% for German.

In German, the fact that TEDx talks sometimes worse recording conditions than TED talks was reflected by the fact that one talk in the German tst2015 set had WERs above 45% and another above 30%, while for all other talks WERs were in the range from 10% to 23%.

Three participants of this year's English ASR track also participated last year. All of them showed significant progress on tst2014, absolute WER improvements ranging from 1.7–5.8 percentage points. This year the lowest WER on tst2014 was 7.1% as compared to 8.4% last year.

Only one participant from this year's German ASR evaluation also participated last year and did not show any progress on tst2014.

3. SLT Track

3.1. Definition

The SLT track required participants to translate the English and German talks of *tst2015* from the audio signal (see Section 2). 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. Furthermore, in contrast to the previous years, this year no manual segmentation into sentences was provided. Therefore, participants needed to develop methods to automatically segment the text and insert punctuation marks.

For German as a source language, participants had to translate into English. For English as source language, participants could choose to translate into one or more languages between Chinese, Czech, French, German, Thai, Vietnamese.

3.2. Evaluation

For the evaluation, participants could choose to either use their own ASR technology, or to use ASR output provided by the conference organizers.

For English, the ASR output provided by the organizers was a single system output from one of the five submissions to the ASR track. For German we also used the a single best scored submissions from a different participant.

The results of the translation had to be submitted in NIST XML format, the same format used in the MT track (see Section 4).

Since the participants needed to segment the input into sentences, the segmentation of the reference and the automatic translation was different. In order to calculate the automatic evaluation metric, we need to realign the sentences of the reference and the automatic translation. This was done by minimizing the WER between the automatic translation and reference as described in [30].

3.3. Submissions

We received 5 primary and 9 contrastive submissions from nine participants, German to English receiving the most submissions.

3.4. Results

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

4. MT Track

4.1. 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.

As already stated in the Introduction, for each translation direction, in-domain training and development data were supplied through the website of the WIT^3 [12], while out-of-domain training data were made available through the workshop's website. With respect to edition 2014 of the evaluation campaign, some of the talks added to the TED repository during the last year have been used to define the new evaluation sets (tst2015), while the remaining talks have been included in the training sets. For reliably assessing progress of MT systems over the years, the evaluation set of edition 2014 (tst2014) were distributed as progressive test set, when available. Development sets are either the same of past editions or have been built upon the same talks; tst2013 sets were included into the list of development sets.

With respect to all the other directions, the $DeEn$ MT task is an exception; in fact, its evaluation sets (tst2014 and tst2015) derive from those prepared for the ASR/SLT tracks, which consist of TEDx talks delivered in German language; therefore, no overlap exists with TED talks involved in other tasks. Both TEDx- and TED-based development sets have been released for this direction.

Table 2 provides statistics on in-domain texts supplied for training and evaluation purposes for each MT task. Texts are pre-processed (tokenization, Chinese and Thai segmentation) with the tools used for setting-up baseline systems (see below). Statistics on most development sets can be found in the overview paper of the 2014 edition [11].

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 [31] was applied to all languages, with the exception of Chinese and Thai; the former was preprocessed by means of the Stanford Chinese Segmenter [32], while the Thai texts were segmented according to the guidelines[1] de-

[1] http://hltshare.fbk.eu/IWSLT2015/InterBEST2009Guidelines-2.pdf

Table 2: Bilingual training and evaluation corpora statistics.

task	data set	sent	tokens En	tokens foreign	talks
$En \leftrightarrow Zh$	train	210k	4.27M	4.02M	1718
	tst2014	1,068	20,3k	20,0k	12
	tst2015	1,080	20,8k	20,7k	12
$En \leftrightarrow Cs$	train	106k	2.09M	1.76M	918
	tst2015	1,080	20,8k	17,9k	12
$En \leftrightarrow Fr$	train	208k	4.23M	4.51M	1711
	tst2014	1,305	24,8k	27,5k	15
	tst2015	1,080	20,8k	22,0k	12
$En \leftrightarrow De$	train	194k	3.94M	3.68M	1597
$\rightarrow$	tst2014	1,305	24,8k	23,8k	15
$\rightarrow$	tst2015	1,080	20,8k	19,7k	12
$\leftarrow$	$\text{tst2014}_{\text{TEDx}}$	1,414	28,1k	27,6k	10
$\leftarrow$	$\text{tst2015}_{\text{TEDx}}$	2,809	41,0k	38.8k	14
$En \leftrightarrow Th$	train	84k	1.66M	2.84M	746
	tst2015	756	15,1k	25,7k	9
$En \leftrightarrow Vi$	train	131k	2.63M	3.32M	1192
	tst2015	1,080	20,8k	24,6k	12

fined at InterBEST 2009.[2]

Translation and lexicalized reordering models were trained on the parallel training data by means of the Moses toolkit; 5-gram LMs with improved Kneser-Ney smoothing were estimated on the target side of the training data with the IRSTLM toolkit [33]. The weights of the log-linear interpolation model were optimized on tst2010 with the MERT procedure provided with Moses.

Reference results from baseline MT systems on evaluation sets have been shared among participants after the Evaluation Period, in order to allow them to assess their scores.

4.2. Evaluation

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

The quality of the translations was measured both automatically, against the human translations created by the TED open translation project, and via human evaluation (Section 5).

Case sensitive scores were calculated for the three automatic standard metrics BLEU, NIST, and TER, as implemented in mteval-v13a.pl[3] and tercom-0.7.25[4], by calling:

- `mteval-v13a.pl -c`
- `java -Dfile.encoding=UTF8 -jar tercom.7.25.jar -N -s`

[2]http://thailang.nectec.or.th/interbest/

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

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

Detokenized texts were passed, since the two scorers apply an internal tokenizer. Before the evaluation, Chinese texts were segmented at char level, keeping non-Chinese strings as they are.

In order to allow participants to evaluate their progresses automatically and in identical conditions, an evaluation server was developed. Participants could submit the translation of any development set to either a REST Webservice or through a GUI on the web, receiving as output the three scores BLEU, NIST and TER computed as above. The core of the evaluation server is a shell script wrapping the `mteval` and `tercom` scorers. The REST service is a PHP script running over Apache HTTP, while the GUI on the web is written in HTML with AJAX code. The evaluation server was utilized by the organizers for the automatic evaluation of the official submissions. After the evaluation period, the evaluation on test sets was enabled to all participants as well.

4.3. Submissions

We received submissions from 11 different sites. The total number of primary runs is 40: 26 on *tst2015* and 14 on *tst2014*; 16 primary runs regard the $EnDe$ pair in either one or the other direction, 10 $EnVi$, 6 $EnFr$, 6 $EnZh$ and 2 $EnCs$; in addition, we were asked to evaluate 33 contrastive runs. No submission were received for Thai.

4.4. Results

The results on the 2015 official test set for each participant are shown in Appendix A.1. Scores of baseline systems developed as described in Section 4.1 are reported as well.

For all language pairs but one, we show the case-sensitive BLEU, NIST and TER scores. The exception is the English to Chinese task, for which character-level scores are given.

On three language pairs out of five (*En-*$\{$*Zh,Cs,Fr*$\}$), too few submissions were received to make general comments; we can just observe that all systems setup by participants outperformed the baselines. The tasks involving German and Vietnamese attracted more attention. On German, which is a language notoriously difficult to process, the better systems largely beat the basic methods featured in the baselines (the BLEU scores of the best ranked runs are higher than baselines by about 50%); the SU MT English-German system deserves to be mentioned since its approach outclasses even the runner-up. On Vietnamese tasks, participant scores vary a lot as well; differently than on German, submitted runs hardly provided higher quality than baselines; in particular, on Vietnamese-to-English direction, none was able to improve the baseline translation: despite a deep analysis, we were unable to find a plausible explanation for this surprising outcome.

In Appendix A.2 the results on the progress test sets *test2014* are shown. For each task, the baseline performance is provided again, together with the score of the best tst2014 run submitted in 2014 edition of the Evaluation Cam-

paign. The latter scores can slightly differ from those officially disclosed last year because they have been recomputed by means of the Evaluation Server. Only tasks involving Chinese, French and German are considered here since Czech and Vietnamese languages were not proposed in edition 2014.

In comparing the 2015 results to the best 2014 submissions, different remarks can be done depending on the language. On Chinese tasks, no improvement is observed with respect to last year, when four participants sent primary runs: likely, the larger number of attendees increases the chance of measuring good scores. If on the English-to-French task the 2014 best system is definitely better than the unique participant to the 2015 edition, in the opposite direction both 2015 runs outperform the 2014 best run: therefore, we can softly state that some progress has been made on French, at least in translating it into English. On the contrary, no doubts that the German 2015 systems (in both directions) definitely improved over the 2014 edition, especially noting that the two best 2014 runs were a Rover combination of some of the other best runs. Therefore, the systems by SU, on English-German, and by RWTH and KIT, for German-English, resulted outstandingly effective.

5. Human Evaluation

Human evaluation was carried out on primary runs submitted by participants to two of the MT TED tasks, namely the MT English-German ($EnDe$) task and MT Vietnamese-English ($ViEn$) task. Following the methodology introduced in 2013, human evaluation was based on *Post-Editing* and systems were ranked according to the HTER (Human-mediated Translation Edit Rate) evaluation metric.

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 functions, and a number of studies [34, 35] demonstrate the usefulness of MT to increase 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 is to adopt a human evaluation framework able to maximize the benefit for 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.[5] Furthermore, the HTER metric [36] - which consists of measuring the minimum edit distance between the MT output and its manually post-edited version (*targeted* reference) - has been shown to correlate quite well with human judgments of MT quality.

[5]All the data produced for human evaluation are publicly available through the WIT^3 repository (wit3.fbk.eu).

Table 3: $EnDe$ task: Post-editing information for each Post-editor. PE effort is estimated with HTER. Scores are given in percentage (%).

PEditor	PE Effort	*std-dev*	Sys TER	*std-dev*
PE 1	22.49	16.44	56.43	20.77
PE 2	42.68	26.51	55.59	20.82
PE 3	29.21	22.18	56.00	20.49
PE 4	27.66	15.50	55.77	21.17
PE 5	22.19	17.62	56.38	20.85

The human evaluation dataset and the collected post-edits are described in Section 5.1, whereas the results of the evaluation are presented in Section 5.2.

5.1. Evaluation Data

The human evaluation (HE) datasets contain around 10,000 words each and include subsets of the 12 TED Talks composing the IWSLT 2015 official test sets. We selected around the initial 56% of each talk for the $EnDe$ HE dataset, and around 45% for the $ViEn$ one.[6] 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 600 segments for $EnDe$ and 500 segments for $ViEn$.

This year we received five primary submissions both for the $EnDe$ task and the $ViEn$ task. For each task, the output of the five systems on the HE set was assigned to five professional translators to be post-edited. To cope with translators' variability, an equal number of outputs from each MT system was assigned randomly to each translator (for all the details about data preparation and post-editing see [11] and Appendix B). The resulting evaluation data for each task consist of five new reference translations for each of the sentences in the HE set. Each one of these five references represents the targeted translation of the system output from which it was derived, and four additional translations are available as well for the evaluation of each MT system.

The main characteristics of the work carried out by post-editors are presented in Tables 3 and 4. 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 sentences post-edited by each single translator.

As we can see from the tables, PE effort is highly variable among post-editors, even though in different proportions depending on the task (from 22.19% to 42.68% for $EnDe$, and

[6]This different percentage is due to the fact that the number of words for each HE dataset was fixed to 10,000 but the Vietnamese source texts contain a higher number of words with respect to English.

Table 4: $ViEn$ task: Post-editing information for each Post-editor. PE effort is estimated with HTER. Scores are given in percentage (%).

PEditor	PE Effort	*std-dev*	Sys TER	*std-dev*
PE 1	37.14	21.25	61.38	20.96
PE 2	40.38	20.46	60.34	20.94
PE 3	44.76	23.57	61.66	21.74
PE 4	46.39	25.71	61.69	21.59
PE 5	38.57	26.64	60.14	20.43

Table 5: $EnDe$ Task: human evaluation results. Scores are given in percentage (%). The system name next to the HTER score indicates the first system in the ranking with respect to which differences are statistically significant at $p < 0.01$.

System Ranking	**HTER** ***HE Set all PErefs***	**HTER** HE Set tgt PEref	**TER** HE Set ref	**TER** Test Set ref
SU	**16.16**$^{\text{UEDIN}}$	21.09	51.15	51.13
UEDIN	**21.84**$^{\text{PJAIT}}$	27.99	56.39	56.05
KIT	**22.67**$^{\text{PJAIT}}$	28.98	55.82	55.52
HDU	**23.42**$^{\text{PJAIT}}$	29.93	57.32	56.94
PJAIT	**28.18**	35.68	59.51	59.03
Rank Corr.		1.00	0.90	0.90

from 37.14% to 46.39% for $ViEn$). Data about weighted standard deviation confirm post-editor variability, showing that translators produced quite different post-editing effort distributions.

To further study post-editors' behaviour, we exploited the official reference translations available for the two MT tasks and we calculated the TER of the MT outputs assigned to each translator for post-editing (*Sys TER* Column in Tables 3 and 4), 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. These results are in line with those observed in IWSLT 2013 and 2014 for different datasets and language pairs.

5.2. Results

The outcomes of the two previous rounds of human evaluation through post-editing [10, 11] demonstrated that HTER computed against all the references produced by all post-editors allow a more reliable and consistent evaluation of MT systems with respect to HTER calculated against the targeted reference only. In light of these findings, also this year systems were officially ranked according to HTER calculated on all the collected post-edits.

Table 6: $ViEn$ Task: human evaluation results. Scores are given in percentage (%). The system name next to the HTER score indicates the first system in the ranking with respect to which differences are statistically significant at $p < 0.01$ (the asterisk indicates significance at $p < 0.05$).

System Ranking	**HTER** ***HE Set all PErefs***	**HTER** HE Set tgt PEref	**TER** HE Set ref	**TER** Test Set ref
JAIST	**32.24**$^{\text{TUT}}$	37.25	60.10	62.35
UMD	**32.71**$^{\text{TUT}}$	37.99	58.92	59.19
PJAIT	**34.27**$^{\text{TUT*}}$	40.50	59.48	62.20
TUT	**38.50**	43.42	62.49	62.69
UNETI	**41.42**	47.97	64.21	66.33
Rank Corr.		1.00	0.70	0.70

Official results and rankings are presented in bold in Tables 5 and 6, which also present HTER scores calculated on the targeted reference only and TER results – both on the HE set and on the full test set – calculated against the official reference translation used for automatic evaluation (see Section 4.2 and Appendix A).[7]

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 [37], a statistical test well-established in the NLP community [38] and that, especially for the purpose of MT evaluation, has been shown [39] to be less prone to type-I errors than the bootstrap method [40]. In this study, the approximate randomization test was based on 10,000 iterations. For the $EnDe$ task, we can see in Table 5 that the top-ranked system (SU) is significantly better than all the other systems, while UEDIN, KIT, and HDU are not significantly different from each other but only with respect to PJAIT. For the $ViEn$ task, Table 6 shows that a winning system cannot be indicated, as there is no system that is significantly better than all other systems; the three top-ranking systems (JAIST, UMD, PJAIT) are significantly better than the two bottom-ranking systems (TUT, UNETI).

Some additional observations can be drawn by comparing HTER and TER results given in the tables, which largely confirm previous years' findings. First, we observe a considerable HTER reduction when using all collected post-edits (*all PErefs*) with respect to both the HTER obtained using the targeted post-edit (*tgt PEref*) and the TER obtained using the independent reference (*ref*). This reduction clearly confirms that exploiting all the available reference translations is a viable way to control and overcome post-editors' variability, giving an HTER which is more informative about the real performances of the systems. Moreover, the correlation between evaluation metrics is measured using *Spearman's rank*

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

correlation coefficient $\rho \in$ [-1.0, 1.0]. We can see from the tables that TER rankings correlate well with the official HTER. Also, the observed shifts in the ranking occur only where the differences between systems are not statistically significant.

To conclude, the post-editing task introduced for manual evaluation brought benefit to the IWSLT community, and in general to the MT field. Indeed, producing post-edited versions of the participating systems' outputs allowed us to carry out a quite informative evaluation which minimizes the variability of post-editors, who naturally tend to diverge from the post-editing guidelines and personalize their translations. Furthermore, a number of additional reference translations are made available to the community for further development and evaluation of MT systems.

6. Conclusions

In this paper, we presented the organisation and outcomes of the 2015 IWSLT Evaluation Campaign. The IWSLT evaluation provides a venue where core technologies for spoken language translation can be evaluated on many different languages and compared not only across research teams but also over time. This year the evaluation was attended by 16 groups – i.e. 6 from Asia, 7 from Europe, and 3 from America. To honor the local organizer of this year, we added among the offered translation directions also English-Vietnamese, which finally attracted several participants. In order to simulate a real subtitling use case, the ASR and SLT tracks were run this year without providing any segmentation of the input speech. Then, in order to improve the automatic evaluation of the MT and SLT tracks, a new evaluation server was developed where participants could submit primary and contrastive runs at any time. Finally, for the two most popular MT runs, a manual evaluation was carried out with professional translators aiming at measuring MT quality in terms of post-editing effort required to fix the MT outputs. Concerning future plans, we are considering to extend the translation task, which now focus on TED talks only, to two other application scenarios: video conferences and lectures.

7. Acknowledgements

The human evaluation and part of the work by FBK's authors were carried out under the CRACKER project, which receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 645357.

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] ——, "Report on the 11th IWSLT Evaluation Campaign, IWSLT 2014," in *Proceedings of the Eleventh International Workshop on Spoken Language Translation (IWSLT 2014)*, Lake Tahoe, USA, 2014.

[12] 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

[13] K. Wolk and K. Marasek, "PJAIT Systems for the IWSLT 2015 Evaluation Campaign Enhanced by Comparable Corpora," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[14] V. T. Hong, H. V. Thuong, V. N. Van, and T. L. Tien, "System description: IWSLT 2015 for Machine Translation," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[15] V. H. Nguyen, Q. B. Nguyen, T. T. Vu, and C. M. Luong, "The IOIT English ASR system for IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[16] H. D. Tran, J. Dennis, and W. Z. Ng, "The I2R ASR System for IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[17] H.-L. Trieu, T.-Q. Dang, P.-T. Nguyen, and L.-M. Nguyen, "The JAIST-UET-MITI Machine Translation Systems for IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[18] M. Heck, Q. T. Do, S. Sakti, G. Neubig, and S. Nakamura, "The NAIST English Speech Recognition System for IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[19] T. Nomura, H. Tsukada, and T. Akiba, "Improvement of Word Alignment Models for Vietnamese-to-English Translation," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[20] J.-T. Peter, F. Toutounchi, S. Peitz, P. Bahar, A. Guta, and H. Ney, "The RWTH Aachen Machine Translation Systems for IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[21] M. Kazi, B. Thompson, E. Salesky, T. Anderson, G. Erdmann, E. Hansen, B. Ore, K. Young, J. Gwinnup, M. Hutt, and C. May, "The MITLL-AFRL IWSLT 2015 MT System," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[22] M. Huck and A. Birch, "The Edinburgh Machine Translation Systems for IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[23] M. Á. D. A. Teba, A. A. M. Villaronga, S. P. Gozalbes, A. G. Pastor, J. A. S. Navarro, J. C. Saiz, and A. J. Císcar, "The MLLP ASR Systems for IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[24] L. Jehl, P. Simianer, J. Hitschler, and S. Riezler, "The Heidelberg University English-German translation system for IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[25] M. G. Martinez, L. Barrault, A. Rousseau, P. Deléglise, and Y. Estève, "The LIUM ASR and SLT Systems for IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[26] A. Axelrod and M. Carpuat, "The UMD Machine Translation Systems at IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[27] T.-L. Ha, J. Niehues, E. Cho, M. Mediani, and A. Waibel, "The KIT Translation Systems for IWSLT 2015," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[28] M. Müller, T. S. Nguyen, M. Sperber, K. Kilgour, S. Stüker, and A. Waibel, "The 2015 KIT IWSLT Speech-to-Text Systems for English and German," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[29] M.-T. Luong and C. D. Manning, "Stanford Neural Machine Translation Systems for Spoken Language Domains," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, 2015.

[30] E. Matusov, G. Leusch, O. Bender, , and H. Ney, "Evaluating Machine Translation Output with Automatic Sentence Segmentation," in *Proceedings of the 2nd International Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, USA, 2005.

[31] 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.

[32] 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.

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

[34] 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

[35] 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.

[36] 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.

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

[38] 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.

[39] 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

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

[41] M. Federico, N. Bertoldi, M. Cettolo, M. Negri, M. Turchi, M. Trombetti, A. Cattelan, A. Farina, D. Lupinetti, A. Martines, A. Massidda, H. Schwenk, L. Barrault, F. Blain, P. Koehn, C. Buck, and U. Germann, "The MateCat Tool," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, August 2014, pp. 129–132. [Online]. Available: http://www.aclweb.org/anthology/C14-2028

Appendix A. Automatic Evaluation

A.1. Official Testset (*tst2015*)

· All the sentence IDs in the IWSLT 2015 testset were used to calculate the automatic scores for each run submission.
· MT systems are ordered according to the *BLEU* metrics.
· *WER*, *BLEU* and *TER* scores are given as percent figures (%).

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

System	WER	(# Errors)
HLT-I2R	7.7	(1,403)
IOIT	13.8	(2,523)
KIT	9.2	(1,689)
NAIST	12.0	(2,197)
MITLL-AFRL	**6.6**	**(1,201)**
MLLP	13.3	(2,421)

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

System	WER	(# Errors)
KIT	20.3	(6,931)
LIUM	**17.6**	**(6,010)**
MLLP	43.3	(14,787)

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

System	*character-based*	
	BLEU	TER
MITLL-AFRL	**18.02**	**75.75**

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

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
LIUM	**18.51**	**79.06**	**20.02**	**76.41**

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

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
KIT	**0.1618**	**78.28**	**16.92**	**76.71**

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

System	*case sensitive*		*case insensitive*	
	BLEU	TER	BLEU	TER
KIT	**19.64**	**62.22**	**20.83**	**60.23**
RWTH	18.79	65.18	20.23	62.62

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

System	*character-based*		
	BLEU	NIST	TER
UEDIN	**25.39**	6.3985	60.83
MITLL-AFRL	24.31	**6.4136**	**59.00**
BASELINE	21.86	5.8640	65.94

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

System	*case sensitive*		
	BLEU	NIST	TER
MITLL-AFRL	**16.86**	**5.2565**	**67.31**
BASELINE	13.59	4.8918	68.01

TED : MT English-Czech (MT$_{EnCs}$)

System	*case sensitive*		
	BLEU	NIST	TER
PJAIT	**17.17**	**5.1056**	**63.00**
BASELINE	14.74	4.7458	65.80

TED : MT Czech-English (MT$_{CsEn}$)

System	*case sensitive*		
	BLEU	NIST	TER
PJAIT	**25.07**	6.4026	55.74
BASELINE	22.44	6.1186	57.99

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

System	*case sensitive*		
	BLEU	NIST	TER
PJAIT	**32.79**	**7.3222**	**49.15**
BASELINE	30.54	6.9957	51.51

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

System	*case sensitive*		
	BLEU	NIST	TER
PJAIT	**32.75**	7.2769	48.41
UMD	32.59	**7.3708**	**47.12**
BASELINE	31.94	7.3415	47.55

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

System	*case sensitive*		
	BLEU	NIST	TER
SU	**30.85**	**6.9898**	**51.13**
KIT	26.18	6.4640	55.52
UEDIN	26.02	6.4518	56.05
HDU	24.96	6.3170	56.94
PJAIT	22.51	6.0412	59.03
BASELINE	20.08	5.7613	61.37

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

System	*case sensitive*		
	BLEU	NIST	TER
RWTH	**31.50**	**7.7932**	**47.11**
KIT	31.08	7.7471	47.24
PJAIT	26.08	7.0350	52.34
BASELINE	21.78	6.4984	55.45

TED : MT English-Vietnamese (MT$_{EnVi}$)

System	*case sensitive*		
	BLEU	NIST	TER
PJAIT	**28.39**	6.6650	56.01
JAIST	28.17	**6.7092**	55.84
KIT	26.60	6.4014	58.26
SU	26.41	6.5986	**55.60**
UNETI	22.93	6.0218	60.33
BASELINE	27.01	6.4716	58.42

TED : MT Vietnamese-English (MT$_{ViEn}$)

System	*case sensitive*		
	BLEU	NIST	TER
PJAIT	**23.46**	5.7314	62.20
UMD	21.57	**5.7831**	**59.19**
JAIST	21.53	5.6413	62.35
UNETI	20.18	5.1443	66.33
TUT	19.78	5.4559	62.69
BASELINE	24.61	5.9259	59.32

A.2. Progress Testset (*tst2014*)

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

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

System	WER	(# Errors)
HLT-I2R	8.9	(1,950)
IOIT	13.9	(3,036)
KIT	9.7	(1,689)
NAIST	10.4	(2,268)
MITLL-AFRL	**7.1**	**(1,549)**
MLLP	19.5	(4,258)

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

System	WER	(# Errors)
KIT	**(24.0)**	**(5,660)**
LIUM	26.5	(6,254)
MLLP	49.4	(11,657)

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

System	*character-based*		
	BLEU	NIST	TER
UEDIN	**19.63**	**5.5483**	68.05
MITLL-AFRL	18.51	5.5294	**66.73**
BASELINE	17.74	5.2514	71.23
BEST IWSLT2014	21.64	5.8732	65.66

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

System	*case sensitive*		
	BLEU	NIST	TER
MITLL-AFRL	**14.14**	**4.6736**	**72.55**
BASELINE	11.43	4.3935	72.65
BEST IWSLT2014	15.63	4.9138	69.67

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

System	*case sensitive*		
	BLEU	NIST	TER
PJAIT	**31.88**	**7.4901**	**47.92**
BASELINE	30.31	7.2488	50.18
BEST IWSLT2014	36.99	7.9127	45.20

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

System	*case sensitive*		
	BLEU	NIST	TER
UMD	**33.20**	**7.4807**	**46.32**
PJAIT	32.92	7.3747	48.25
BASELINE	32.20	7.3677	47.60

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

System	*case sensitive*		
	BLEU	NIST	TER
SU	**27.58**	**6.8218**	**52.50**
UEDIN	24.01	6.3821	57.04
KIT	23.31	6.4106	56.51
HDU	23.22	6.2500	57.81
PJAIT	20.68	5.9978	59.78
BASELINE	18.49	5.7409	61.66
BEST IWSLT2014	23.25	6.3415	57.27

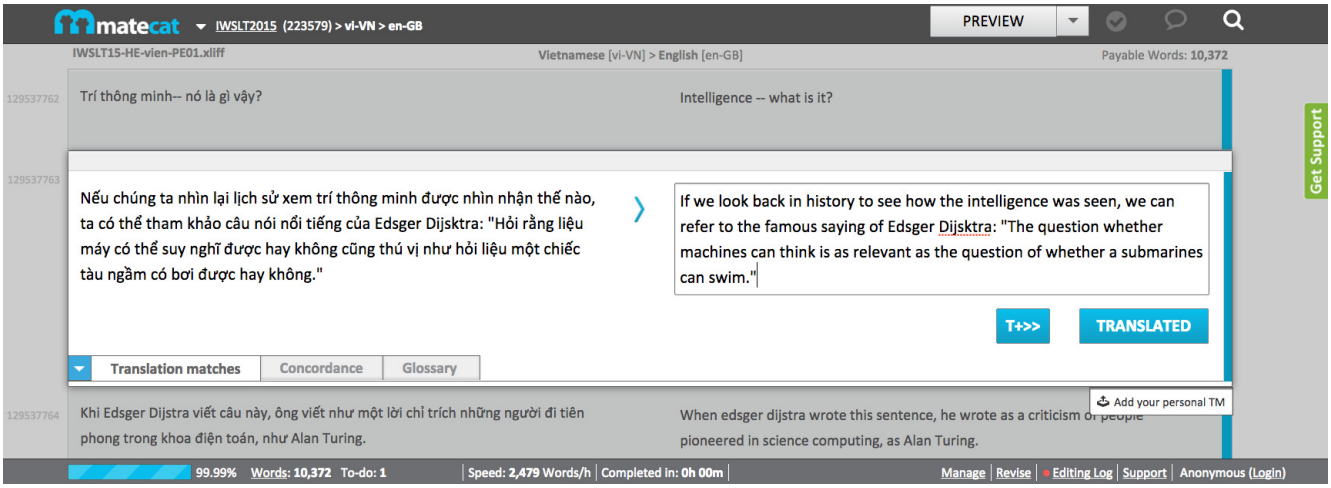
TEDX : MT German-English (MT$_{DeEn}$)

System	*case sensitive*		
	BLEU	NIST	TER
RWTH	**26.18**	**6.7160**	**55.15**
KIT	25.18	6.5795	55.76
PJAIT	21.92	6.0407	60.59
BASELINE	17.99	5.5186	64.36
BEST IWSLT2014	25.80	6.7011	55.07

Appendix B. Human Evaluation

Interface used for the bilingual post-editing task

Post-editing was carried out using MateCat[8] [41], which is a web-based open-source professional CAT tool developed within the EU funded project Matecat.



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. Also, depending on the style of the source language talk, you can use the corresponding style in the target language (*e.g.* if the talk uses a friendly/colloquial style you can use informal words too).

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.

The document you have to post-edit is composed of around the first half of 12 different talks. Below you can find the name of the speaker and the title of each talk.

1. Alex Wissner-Gross: A new equation for intelligence.
2. Ash Beckham: We're all hiding something let's find the courage to open up.
3. Mary Lou Jepsen: Could future devices read images from our brains?
4. Ziauddin Yousafzai: My daughter Malala.
5. Geena Rocero: Why I must come out.
6. Kevin Briggs: The bridge between suicide and life.
7. Chris Kluwe: How augmented reality will change sports and build empathy.
8. Stella Young: I'm not your inspiration thank you very much.
9. Zak Ebrahim: I am the son of a terrorist here's how I chose peace.
10. David Chalmers: How do you explain consciousness.
11. Meaghan Ramsey: Why thinking you're ugly is bad for you.
12. Marc Kushner: Why the buildings of the future will be shaped by you.

[8] www.matecat.com

The RWTH Aachen German to English MT System for IWSLT 2015

Jan-Thorsten Peter, Farzad Toutounchi, Stephan Peitz, Parnia Bahar, 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 of the *International Workshop on Spoken Language Translation* (IWSLT) 2015. We participated in the MT and SLT tracks for the German→English language pair. We employ our state-of-the-art phrase-based and hierarchical phrase-based baseline systems for the MT track. The phrase-based system is augmented with joint translation and reordering model and maximum expected BLEU training for phrasal, lexical and reordering models. Furthermore, we apply feed-forward and recurrent neural language and translation models for reranking. We also train attention-based neural network models and utilize them in reranking the *n*-best lists for both phrase-based and hierarchical setups. On top of all our systems, we use system combination to enhance the translation quality by combining individually trained systems. In the SLT track, we additionally perform punctuation prediction on the automatic transcriptions employing hierarchical phrase-based translation.

1. Introduction

We describe the statistical machine translation (SMT) systems developed by RWTH Aachen University for the evaluation campaign of IWSLT 2015. We participated in the machine translation (MT) track and the spoken language translation (SLT) track for the German→English language pair. A combination of several single machine translation engines has proven to be highly effective on previous joint submission, e.g. [1, 2], and a similar approach is used for this task. We train individual systems using state-of-the-art phrase-based and hierarchical phrase-based translation engines. Each single system is a pipeline including either a phrase-based or a hierarchical decoder with additional models such as hierarchical reordering models, word class (cluster) language models, joint translation and reordering models, discriminative phrase training and reranking with different neural network models. For the spoken language translation task, the ASR output is enriched with punctuation and case information. 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 joint translation and reordering models and discriminative phrase training, and sections 2.6, 2.7, and 2.8 describe the neural network models used in our translation systems, which are very effective in the shared task. Sections 2.9 explains the system combination pipeline applied on the individual systems for obtaining the combined system. Our experiments for each track are summarized in Section 3 and we conclude with Section 4.

2. SMT Systems

For the IWSLT 2015 evaluation campaign, RWTH utilizes state-of-the-art phrase-based and hierarchical translation systems. GIZA++ [3] is employed to train word alignments. We used *MultEval* [4] to evaluate our systems on the BLEU [5] and TER [6] measures. Due to using *MultEval*, BLEU scores are case-sensitive and TER scores are case-insensitive.

2.1. Phrase-based Systems

Our phrase-based decoder is the implementation of the *source cardinality synchronous search* (SCSS) procedure described in [7] in RWTH's open-source SMT toolkit, Jane 2.3[1] [8], 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 enhanced low frequency feature [9]. The parameter weights are optimized with MERT [10] towards the BLEU metric. Additionally, we make use of a hierarchical reordering model (HRM) [11], a high-order word class language model (wcLM) [12], a joint translation and reordering model (cf. Section 2.4), a maximum expected BLEU training scheme (cf. Section 2.5) and reranking with different neural network models (cf. Sections 2.6, 2.7 and 2.8).

[1] http://www-i6.informatik.rwth-aachen.de/jane/

2.2. Hierarchical Phrase-based System

For our hierarchical setups, we also employ the open source translation toolkit Jane 2.3 [13]. In hierarchical phrase-based translation [14], 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 hierarchical phrases, glue rule, and rules with non-terminals at the boundaries, enhanced low frequency feature and *n*-gram language models. We utilize the cube pruning algorithm [15] for decoding. Reranking the *n*-best lists using neural network models is also employed for our hierarchical systems.

2.3. Backoff Language Models

Both phrase-based and hierarchical translation systems use three backoff language models that are estimated with the KenLM toolkit [16] 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, French Shuffled News and both the English and French Gigaword corpora by the cross-entropy difference criterion described in [17]. The selection is then concatenated with all available remaining monolingual data and used to build an unpruned language model. The in-domain language model is 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 [12] to compute the wcLM on the same data as the general-domain LM.

2.4. Joint Translation and Reordering Models in Phrase-Based System

Joint translation and reordering (JTR) model [18] is introduced into the log-linear framework of our phrase-based system in order to include lexical and reordering dependencies beyond phrase-boundaries. The JTR model allows for more context than the previously developed extended translation model [19]. The unique JTR sequences are obtained by converting the full bilingual data and the corresponding Viterbi alignments. We train count-based 7-gram models with modified Kneser-Ney smoothing [20] on the JTR sequences using the KenLM toolkit [16].

In order to have the necessary information about the JTR sequences available during decoding, we annotate each phrase-table entry with the corresponding JTR sequence. Within the phrase-based decoder, we extend each search state such that it additionally stores the JTR model history. During decoding, a reordering token has to be appended to the beginning of the hypothesized JTR sequence, if the alignment step from the previous JTR token in the history to the current token is non-monotone.

Including the JTR model improved our phrase-based baseline system by 0.7 BLEU on tst2013.

2.5. 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 two types of features under a maximum expected BLEU objective [21]. We used the TED portion of the data for discriminative training, since it is high quality in-domain data of reasonable size. This makes training feasible while at the same time providing an implicit domain adaptation effect. For our gradient based update method we generate 100-best lists on the training data which are used as training samples similar to [21]. A leave-one-out heuristic [22] is applied to circumvent over-fitting. Here, we follow an approach similar to [23]. Each feature type is first discriminatively trained, then condensed into a single feature for the log-linear model combination and finally optimized with MERT. We simultaneously train phrase pair features and phrase-internal word pair features, adding two models to the log-linear combination. In the tables in Section 3 we denote the maximum expected BLEU training as *MaxExpBleu*.

2.6. Feed-Forward Neural Network Models

We use four feed-forward neural network (FFNN) models with similar structure as the models used by [24, 25]. The models and following neural network models are applied for reranking 1000-best lists. The new weights are trained with one additional MERT iteration.

All networks are trained with different input features or layers:

- Language model (LM), the 7 last words on the target side, with two hidden layers (1000 and 500 nodes)
- Joint model (JM), the 5 source words around the aligned source word (2 before the aligned word, and 2 after it) and the 4 last words on the target side, with two hidden layers (1000 and 500 nodes)
- Translation model (TM), the 5 source words around the aligned source word, with two hidden layers (1000 and 500 nodes)
- Translation model (TM), the 5 source words around the aligned source word, with three hidden layers (2000, 2000, and 1000 nodes)

The output layer in all cases is a softmax layer with a short list of 10000. All remaining words are clustered into 1000 classes, and the corresponding class probabilities are predicted. The neural network was implemented using Theano [26, 27].

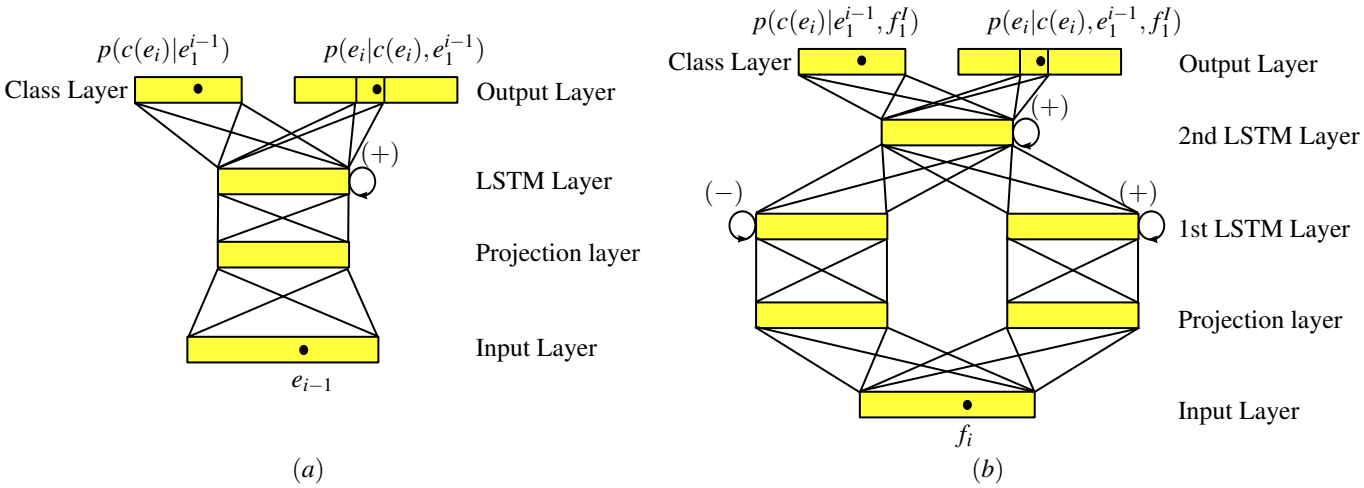


Figure 1: Architecture of the deep recurrent (a) language model, and (b) bidirectional translation model. By $(+)$ and $(-)$, we indicate a processing in forward and backward time directions, respectively. A single source projection matrix is used for the forward and backward branches.

2.7. Recurrent Neural Network Models

Our systems apply reranking on 1000-best lists using recurrent language and translation models. The recurrency is handled with the long short-term memory (LSTM) architecture [28] and we use a class-factored output layer for increased efficiency as described in [29]. All neural networks are trained using 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 [30]. This requires a one-to-one word alignment, which is generated by introducing 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. Two language models are used for reranking, one is trained on the in-domain data, and the other on the entire monolingual data. The in-domain language model is set up with 300 nodes in both the projection and the hidden LSTM layer, while the general-domain language model is set up with 500 nodes in both layers. The general-domain language model is the same model which was used in the IWSLT 2014 evaluations [31]. For the BTM, the in-domain bilingual data is used for training. Furthermore, 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 LM and BTM networks are shown in Figure 1. The neural network was implemented using the RWTHLM toolkit.[2]

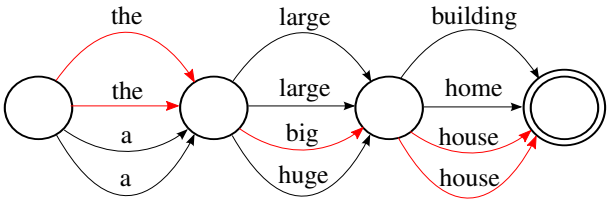


Figure 2: System A: *the large building*; System B: *the large home*; System C: *a big house*; System D: *a huge house*; Reference: *the big house*.

2.8. Attention Based Recurrent Neural Network

The neural network models described in Section 2.6 and Section 2.7 are either used as pure language models or rely on the alignments given by the underlying system. To avoid this dependency on the alignment while maintaining the translation model we also use an attention-based recurrent neural network model as proposed in [32]. The model uses gated recurrent units as proposed by [33]. They have comparable properties to the LSTM architecture used by the recurrent neural networks in Section 2.7. We use a bidirectional layer on the source side with 1000 nodes for each direction and a unidirectional model with 1000 nodes for the target side. The GroundHog toolkit[3] was used to train two models, one on the in-domain data and one on the full data.

2.9. System Combination

System combination is applied to produce consensus translations from multiple hypotheses which are obtained from different translation approaches. The consensus translations outperform the individual hypotheses in terms of translation quality. A system combination implementation which has been developed at RWTH Aachen University [34] is used to

[2] https://www-i6.informatik.rwth-aachen.de/web/Software/rwthlm.php

[3] https://github.com/lisa-groundhog/GroundHog

combine the outputs of different engines.

The first step in system combination is generation of confusion networks (CN) from I input translation hypotheses. We need pairwise alignments between the input hypotheses, and the alignments are obtained by METEOR [35]. The hypotheses are then reordered to match a selected skeleton hypothesis in terms of word ordering. We generate I different CNs, each having one of the input systems as the skeleton hypothesis, and the final lattice will be the union of all I generated CNs. In Figure 2 an example of a confusion network with $I = 4$ input translations is depicted. The decoding of a confusion network is finding the shortest path in the network. Each arc is assigned a score of a linear model combination of M different models, which include word penalty, 3-gram language model trained on the input hypotheses, a binary primary system feature that marks the primary hypothesis, and a binary voting feature for each system. The binary voting feature for a system is 1 iff the decoded word is from that system, otherwise 0. The different model weights for system combination are trained with MERT.

3. Experimental Evaluation

3.1. Machine Translation (MT) Track

For the German→English machine translation task, the word alignment is trained with GIZA++ and we apply the phrase-based decoder, as well as the hierarchical phrase-based decoder implemented in Jane. We use all permissible parallel data for the IWSLT 2015 systems in training the translation model. In a preprocessing step the German source is decompounded [36] and part-of-speech-based long-range verb reordering rules [37] are applied. 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 addition, we tune our systems on the development set `dev2012`, which contains manual transcriptions from German talks and is more similar to the evaluation data. As `tst2013` is also a manual transcription of TED talks, we will focus on the results for the `dev2012`-tuned system on this evaluation data set. The performance of the individual MT systems based on phrase-based and hierarchical phrase-based decoders is summarized in Table 1.

The phrase-based baseline reaches a performance of 28.0 BLEU on `tst2013`. Adding the joint translation and reordering (JTR) models to baseline increases the BLEU scores to 28.7 on `tst2013`. Introducing maximum expected BLEU training on top of JTR improves the translation quality by 0.5 BLEU on `tst2013`. We also apply different neural network models for reranking the 1000-best lists obtained by phrase-based system which is augmented with JTR. We use the four feed-forward models described in Section 2.6, and they each improve the JTR system by 0.1 to 0.3 BLEU. Moreover, we employ recurrent models described in Section 2.7, and depending on the model they can also improve the performance by up to 0.4 BLEU. Introducing the attention-based recurrent model (cf. Section 2.8), enhances the translation quality of the phrase-based system with JTR by 0.8 BLEU. So far all the neural network models were applied individually. In the last two rows of the phrase-based section in Table 1, we use all the above neural networks simultaneously for reranking the n-best lists of the phrase-based system including JTR, and we improve the translation quality by 1.1 and 1.2 BLEU on `tst2013` in two different optimization runs.

The hierarchical baseline system reaches a performance of 28.8 BLEU on `tst2013`. We tried to add source reordering to the hierarchical baseline. Athough it does not improve the translation quality of `tst2013`, we keep it as an individual system for our system combination pipeline. Applying a feed-forward neural network language model and a recurrent neural network language model for reranking the 1000-best lists obtained by hierarchical baseline system improves the translation quality by 0.1 and 0.2 BLEU, respectively. We also use the attention-based recurrent neural network in reranking, and it boosts the BLEU scores by 1 and 1.2 points in two different optimization runs. Using attention-based networks trained on the in-domain data also enhances the translation quality of baseline by 0.5 BLEU. Furthermore, we use all the above neural networks at the same time for reranking the n-best lists of the hierarchical baseline system, and the improvement on `tst2013` is 1.1 BLEU.

The final submission system for the MT track of IWSLT 2015 German→English task is the combination of all single systems in Table 1 using the methods described in Section 2.9. In total, 20 systems are combined, and the parameters are tuned on `dev2012`. The performance of the combined system is summarized in Table 2. Comparing to our 2014 submission system, we have an improvement of 1.2 BLEU on `tst2014`.

3.2. Spoken Language Translation (SLT) Track

RWTH participated in the German→English SLT task. Punctuation marks and case information are reintroduced by applying a monolingual hierarchical phrase-based translation system as described in [38]. In such a system, hierarchical phrases with a maximum of one non-terminal symbol are extracted and the feature weights can be tuned with MERT. In addition, we add a word class language model (wcLM) to the log-linear model combination.

Table 3 shows a comparison of monolingual phrase-based [39] and hierarchical translation systems tuned on different optimization criteria.

For this task, tuning a monolingual hierarchical translation system on BLEU seems to work better than optimizing towards F_2-Score. In any case it outperforms the phrase-based system. Furthermore, applying a word class language model (wcLM) seems to help as well in terms of BLEU and TER.

Since punctuation prediction and recasing are applied before the actual translation, our translation system can be kept

Table 1: Results of the individual systems for the German→English MT task. BLEU scores are case-sensitive and TER scores are case-insensitive.

	dev2012		tst2010		tst2011		tst2012		tst2013	
Individual Systems	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
SCSS Baseline	25.3	59.6	29.8	48.8	35.4	43.4	29.7	48.9	28.0	51.1
+ **JTR model**	26.4	58.6	30.0	48.4	36.2	42.6	30.3	48.1	28.7	50.3
+ MaxExpBleu	26.8	57.7	30.9	47.2	37.0	41.8	30.7	47.5	29.2	49.9
+ FFNN-LM	26.6	58.3	30.4	48.4	36.6	42.4	30.6	48.3	29.0	50.1
+ FFNN-JM	26.7	58.2	30.1	48.4	36.3	42.4	30.7	48.0	28.8	50.4
+ FFNN-TM	26.7	58.2	30.1	48.4	36.4	42.2	30.4	48.1	28.9	50.1
+ FFNN-TM*	26.4	58.3	31.3	47.6	37.4	41.7	31.6	47.3	29.0	50.1
+ RNN-LM	26.8	58.3	30.0	48.4	36.1	42.6	30.4	48.3	29.1	50.0
+ RNN-LM-InDomain	26.3	58.4	30.4	48.3	36.6	42.3	30.5	48.1	28.2	50.9
+ RNN-BTM	26.7	57.8	30.8	47.7	37.3	41.7	31.2	47.2	29.1	49.9
+ RNN-Attention	27.0	57.9	31.5	47.1	38.0	41.2	31.8	46.8	29.5	49.6
+ AllAboveNNs	27.4	57.1	31.2	47.2	36.7	42.0	31.6	47.2	29.9	49.0
+ AllAboveNNs†	27.9	56.5	31.8	46.5	37.6	41.1	31.5	46.7	29.8	48.9
Hierarchical Baseline	25.3	60.0	30.2	49.3	35.3	44.0	30.1	49.0	28.8	51.6
+ SrcReordering	25.7	59.2	30.0	49.1	35.7	43.6	30.0	48.9	28.4	51.1
+ FFNN-LM	25.4	60.3	30.1	49.4	35.5	43.8	30.0	49.3	28.9	51.7
+ RNN-LM	25.9	60.0	29.9	49.4	35.2	43.7	30.1	49.2	29.0	51.4
+ RNN-Attention	26.4	59.3	31.4	48.2	36.5	42.9	31.4	47.8	30.0	50.5
+ RNN-Attention†	26.4	59.3	30.6	48.9	35.8	43.6	30.6	48.6	29.8	50.6
+ RNN-Attention-InDomain	26.3	59.0	30.8	48.5	36.0	43.2	30.8	48.3	29.3	50.9
+ AllAboveNNs	26.7	58.6	31.9	47.6	36.9	42.4	31.8	47.3	29.9	50.4

* This FFNN-TM has three hidden layers. The other FFNNs have two hidden layers. (cf. Section 2.6)

† A different optimization run.

Table 2: Results of the combined system for the German→English MT task submission. `tst2014` and `tst2015` results are computed by the task organizers. BLEU scores are case-sensitive and TER scores are case-insensitive.

	dev2012		tst2013		tst2014		tst2015	
System	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
Best Individual System	27.4	57.1	29.9	49.0	25.2	56.4	31.1	48.3
Combined System (2015 Submission)	28.2	57.0	30.5	49.0	26.2	55.2	31.5	47.1
2014 Submission	27.0	57.2	27.6	52.1	25.0	55.5	-	-

Table 3: Results of the German→English SLT task. Scores for `tst2015` (case-sensitive) are computed by the task organizers.

	Prediction	**Optimization**	dev2012		tst2013		tst2015	
System	**Method**	**Criterion**	BLEU	TER	BLEU	TER	BLEU	TER
SCSS Baseline	phrase-based	F_2	20.5	62.6	18.6	63.7	-	-
		BLEU	20.0	65.1	18.4	65.8	-	-
	hierarchical	F_2	20.9	62.1	18.7	63.4	-	-
		BLEU	20.9	62.5	19.0	63.4	-	-
	+ wcLM	BLEU	21.3	61.7	19.1	62.8	-	-
+ AllAboveNNs	+ wcLM	BLEU	21.6	61.1	19.8	62.4	18.8	65.2

completely unchanged and we are able to use our final system from the MT track directly. We use *SCSS Baseline + AllAboveNNs* (cf. Table 1) for our final submission.

4. Conclusion

RWTH participated in the MT and SLT tracks for the German→English IWSLT 2015 evaluation campaign.

The baseline systems for the MT track utilize our state-of-the-art phrase-based and hierarchical translation decoders and we were able to improve them by applying maximum expexted BLEU training and employing several neural network models for reranking the *n*-best lists. We built several single machine translation engines which are based on either phrase-based or hierarchical decoders, and combined all the built systems using our system combination pipeline. We achieve a performance of 26.2 in BLEU and 55.2 in TER for `tst2014` and 31.5 in BLEU and 47.1 in TER for `tst2015`, and we improve the BLEU scores by 1.2 point on the `tst2014` compared to our 2014 system.

For the SLT track, the ASR output was enriched with punctuation and casing information by a hierarchical translation system.

5. Acknowledgements

This paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 645452 (QT21).

6. References

[1] 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. Cetolo, 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.

[2] 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.

[3] 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.

[4] J. H. Clark, C. Dyer, A. Lavie, and N. A. 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: Human Language Technologies: Short Papers - Volume 2*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 176–181.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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, WA, USA, Oct. 2013, pp. 1377–1381.

[13] 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.

[14] D. Chiang, "Hierarchical Phrase-Based Translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[15] 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.

[16] 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.

[17] R. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *ACL (Short Papers)*, Uppsala, Sweden, July 2010, pp. 220–224.

[18] A. Guta, T. Alkhouli, J.-T. Peter, J. Wuebker, and H. Ney, "A Comparison between Count and Neural Network Models Based on Joint Translation and Reordering Sequences," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015.

[19] A. Guta, J. Wuebker, M. Graça, Y. Kim, and H. Ney, "Extended Translation Models in Phrase-based Decoding," in *Proceedings of the EMNLP 2015 Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal, Sept. 2015.

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

[21] 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.

[22] 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.

[23] 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.

[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 (Volume 1: Long Papers)*, Baltimore, Maryland, June 2014, pp. 1370–1380.

[25] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous Space Translation Models with Neural Networks," Montréal, Canada, June 2012, pp. 39–48.

[26] 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, oral Presentation.

[27] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM Neural Networks for Language Modeling," in *Interspeech*, Portland, OR, USA, Sept. 2012.

[30] 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.

[31] J. Wuebker, S. Peitz, A. Guta, and H. Ney, "The RWTH Aachen Machine Translation Systems for IWSLT 2014," in *International Workshop on Spoken Language Translation*, Lake Tahoe, CA, USA, Dec. 2014.

[32] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, May 2015.

[33] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, October 2014, pp. 1724–1734.

[34] 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), Gothenberg, Sweden, Apr. 2014, pp. 29–32.

[35] 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, June 2005, pp. 65–72.

[36] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *Proceedings of European Chapter of the ACL (EACL 2003)*, 2003, pp. 187–194.

[37] 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.

[38] 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, pp. 271–278.

[39] 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.

The MITLL-AFRL IWSLT 2015 Systems[†]

Michaeel Kazi[1], Brian Thompson[1], Elizabeth Salesky[1], Timothy Anderson[2], Grant Erdmann[2], Eric Hansen[2], Brian Ore[2], Katherine Young[2], Jeremy Gwinnup[2], Michael Hutt[2], Christina May[2]

[1]MIT Lincoln Laboratory
Human Language Technology Group
244 Wood Street
Lexington, MA 0220, USA
{first.last}@ll.mit.edu

[2]Air Force Research Laboratory
Human Effectiveness Directorate
2255 H Street
Wright-Patterson AFB, OH 45433
{first.last.*}@us.af.mil

Abstract

This report summarizes the MITLL-AFRL MT, ASR and SLT systems and the experiments run using them during the 2015 IWSLT evaluation campaign. We build on the progress made last year, and additionally experimented with neural MT, unknown word processing, and system combination. We applied these techniques to translating Chinese to English and English to Chinese. ASR systems are also improved by refining improvements developed last year. Finally, we combine our ASR and MT systems to produce a English to Chinese SLT system.

1. Introduction

During the evaluation campaign for the 2015 International Workshop on Spoken Language Translation (IWSLT '15) [1] our experimental efforts in machine translation (MT) centered on 1) the addition of hierarchical decoding systems 2) reranking n-best lists with a neural net encoder-decoder 3) post-processing of unknown words in translation output and 4) system combination.

Experimental efforts for the automatic speech recognition (ASR) task focused on using cutting edge neural net techniques and the combination of HTK and Kaldi-based ASR systems.

We combine both efforts to produce a system for the spoken language translation (SLT) task. Various segmentation and punctuation strategies were explored.

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. Finally, our work on the SLT task is discussed in Section 4.

[†]This material is based upon work supported by the Air Force Research Laboratory under Air Force Contract No. (FA8721-05-C-0002 and/or FA8702-15-D-0001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Air Force Research Laboratory.

2. Machine Translation

2.1. Data Usage

Unless otherwise noted, data described in this section originates from the WMT15 website[1]. We used the parallel in-domain data supplied by WIT3 [2]. In Chinese–English, we additionally used the Yandex corpus[2], Common Crawl, Wiki Headlines, News Crawl, and the LDC Gigaword corpus as sources of monolingual English data for language model training. In English–Chinese we utilized the Chinese portion of the MultiUN corpus as an additional source of language model training data.

2.2. Data Preprocessing and Cleanup

As in past years, we applied a cleaning process to the training data as previously described in [3]. Chinese was segmented with the Stanford Segmenter [4] using both Chinese Treebank (CTB) and Peking University (PKU) models.

2.3. Training

2.3.1. Phrase and Rule Table Training

We used the default Moses scripts when training phrase and rule tables. For Chinese to English, we increased the size of the training corpus by concatenating output from both the CTB and PKU segmentation models while simply repeating the English portion of the corpus. This allows us to extract phrases for a greater number of phrases than one segmentation alone. We also experimented with outputting the k-best segmentation choices for a model while repeating the English portion.

Phrase and rule tables are trained with default Moses scripts or our custom MT pipeline driver. Good-Turing smoothing[5] was applied to both rule and phrase tables.

[1]http://www.statmt.org/wmt15/translation-task.html
[2]https://translate.yandex.ru/corpus?lang=en

2.3.2. Language Model Training

We reuse our BigLM15 from our WMT15 shared translation task submission[6] as our main English machine translation language model. The English data sources listed in Section 2.1 were used to train a very large 6-gram language model. For Chinese, we take a similar approach to English, using the TED in-domain parallel training data and the Chinese portion of the MultiUN corpus. `kenlm` [7] was used to train 6-gram models in both languages. These models were then binarized and stored on local solid-state disks for each machine in our cluster to improve load time and reduce fileserver traffic.

2.4. Baseline MT System

Our system implements a fairly standard statistical machine translation (SMT) architecture. It consists of the following:

- Moses phrase-based [8] or hierarchical decoding with the incremental-search algorithm [9]
- Stanford Chinese character segmentation [4]
- Hierarchical **mslr** lexical reordering [10] for phrase-based systems
- Minimal phrase table [11]
- 7-gram brown-cluster language model with 80 classes
- BigLM15 [6] for English, consisting of WMT newscrawl data, europarl, news commentary
- Drem optimization [12]
- Recurrent neural net language model (RNNLM) rescoring [13]

2.5. Neural MT methods

2.5.1. Chinese to English

We reranked our n-best lists using an end-to-end neural MT system: our own in-house Torch7 [14] implementation of Sutskever et al [15]'s LSTM encoder-decoder approach. This system was trained using varying amounts of out-of-domain UN data, followed by training on TED data. In the following table, the UN sentences were ranked according to bilingual cross-entropy difference [16] (using RNNLM for the language model component) and the top N were chosen to pretrain the network. Once validation error settled down, the networks were then trained over the 200,000 TED training examples. The different models that have been trained can be seen in Table 1. In common among all of them were vocab selection: vocab entries were taken if they appeared at least 10 times in TED, or 100 times in UN, or 5 times in TED and 20 times in UN. Reranking results for individual systems, as well as pairwise combinations, can be seen in Tables 1 and 2.5.1.

Using the best scoring encoder-decoder (#3), and the best scoring combination, we were able to rerank the n-best list from our best hierarchical moses system, and achieved significant gains. In particular, we increased the score from 16.94 to 17.60 cased BLEU on tst2013 for our best hierarchical system (see Table 4).

id	d	N	dev ppl	BLEU
0	4	2.5M	27.15	16.89
1	1	1M	30.01	16.68
2	2	5M	25.54	16.92
3	2	1M	28.26	16.97
4	1	2.5M	27.98	16.62
5	2	2.5M	24.21	16.73

Table 1: Perplexity on `dev2010` (network validation cost), and cased BLEU on `tst2013`. d = LSTM depth and N = cross-entropy filter size for UN data.

	0	2	3	4	5
0	-	16.90	16.89	16.77	16.93
2		-	17.00	16.69	16.96
3			-	16.75	16.89
4				-	16.84
5					-

Table 2: Cased BLEU on `tst2013` for combinations of encoder-decoders, numbered as in Table 1.

2.5.2. English to Chinese

For the English to Chinese task, we achieved gains by using the Neural Network Joint Model (NNJM [17]), and additionally by reranking using RNNsearch [18], the Montreal LISA-lab attention model system. The NNJM was trained using our own in-house implementation in Theano [19]. We integrated NNJM decoding into Moses as a feature function, utilizing self-normalization and precomputation to allow reasonable runtimes. This gave us a gain of approximately 0.2 BLEU over a strong baseline including factored models and RNN rescoring. For RNNsearch, we used GroundHog[3] to train a model, and to compute scores over an n-best list produced by our conventional MT systems. The network sizes used were GroundHog defaults. Like our Chinese to English Torch system, the GroundHog system used MultiUN data in the same way. This gave us an additional 0.4 BLEU gain. A summary of results can be seen in Table 3.

2.6. System combination

This year we experimented with system combination techniques based on Rosti et al [20], a well established technique in machine translation. Our only additional

[3] `https://github.com/lisa-groundhog/GroundHog`

En-Zh System	char BLEU
Baseline	20.37
+ 400 Class Factored LM	20.52
+ RNNLM	21.23
+ NNJM	21.42
+ GroundHog 2M	21.64
-/+ GroundHog 4M	21.85

Table 3: English–Chinese system additions

id	Description	BLEU
0	Hiero, 6-iter Drem Dev10, CLM	16.50
1	(0) + fixed wide beam	16.88
2	(0) + bigdev	16.94
3	(2) + enc-dec	17.60
4	(2) + 3-iter Drem variation	16.60
5	(2) + 6-iter Drem variation	16.66
6	Hiero Incsearch, bigdev, no rescoring	15.58
7	PB, bigdev, no rescoring	14.38
8	PB, ted+nyt CLMs, enc-dec	17.06

Table 4: Some notes on the systems: CLM = brown cluster language model, bigdev = `dev2010 + tst2010 + tst2011 + tst2012`. TED factored LM has 80 classes, nyt (LDC English Gigaword) has 600. All used a variation of LMs trained on WMT'15 data.

contribution was in sub-selecting systems with which to perform system combination. Among our different collaborators, we managed to produce over 30 systems with 400+ decode outputs. With the goal of choosing only 9, we first filtered out systems with scores less than some minimum acceptable value (in our case, 16.50 cased BLEU on `tst2013`). Then, we constructed a distance metric as $1-\text{BLEU}(x, y)$ and performed k-medoids clustering to choose systems that were sufficiently different from each other.

Table 4 lists different systems used for combination, and Table 5 lists a sampling of combinations tried and their case-sensitive BLEU scores on `tst2013`.

Combo id	Systems Used	tst2013 BLEU
0	0+1+2+4+8	17.62
1	0+1+3+5+8	17.64
2	0+1+2+4+5+8	17.64
3	0+1+3+4	17.66
4	0+1+5+8	17.74

Table 5: Top 5 systems out of system combination

2.7. Unknown Word Processing

As in our WMT15 submission [6], we employed unknown word post-processing to handle any unknown words in the translation instead of simply dropping these words. To test the effectiveness of this approach, we decode all test sets where references are available with a bare-bones Moses hierarchical decoding system where no rescoring features are employed. The resulting gains measured in uncased BLEU are shown in Table 6. We note that the improvements in BLEU are smaller for the Chinese–English language pair when compared to our efforts in processing unknown words in other language pairs, such as Russian–English[6], but we feel that employing these processes are still worthwhile due to the positive impact on readability of the machine translation output. Our technique adapted to Chinese–English is described in the following section.

2.7.1. Chinese to English post-processing

The named entity list used for named entity post-processing comes from manual translations of named entities found in train 2014. It was expanded by adding versions of the Chinese name with the common nouns stripped off. A list of 29 typical common nouns endings of named entity phrases was compiled. Common nouns like: **病** (disease), **县** (county), **族** (race/people), **实验室** (laboratory), **湖** (lake), **集团** (corporation), **群岛** (archipelago) can sometimes be optionally included or omitted by the speaker or optionally split off of entities by word segmenters or named entity taggers.

The output is searched for words containing any Chinese characters. Any unknown word consisting of a single character is deleted since single-character entities are rare in this domain (and segmentation errors are a more common explanation for unknown single-character words). If the word is not found in the named entity word list, the list is searched again for the entity with common nouns stripped. Remaining unknown words are deleted from the output.

Test Set	base. BLEU	post. BLEU	Δ BLEU
`tst2013`	16.09	16.19	+0.10
`tst2012`	13.64	13.65	+0.01
`tst2011`	15.21	15.29	+0.08
`tst2010`	12.43	12.50	+0.07

Table 6: NE post-processing improvement measured in uncased BLEU.

2.8. Submission

Our primary Chinese–English MT submission is system #3 in Table 4. We submitted system #4 in Table 5 as contrastive. For English–Chinese, the primary system

was the last entry in Table 3.

These systems were used to decode the `tst2014` and `tst2015` test sets. Results from scoring performed by the workshop organizers are listed in Table 7 including baseline system scores as determined by the organizers.

System	Lang Pair	Test Set	BLEU
Baseline	Zh-En	tst2014	11.43
Primary	Zh-En	tst2014	14.13
Contrastive	Zh-En	tst2014	13.35
Baseline	Zh-En	tst2015	13.59
Primary	Zh-En	tst2015	16.86
Contrastive	Zh-En	tst2015	15.05
Baseline	En-Zh	tst2014	17.74
Primary	En-Zh	tst2014	18.51
Baseline	En-Zh	tst2015	21.86
Primary	En-Zh	tst2015	24.31

Table 7: Official results measured in cased BLEU.

3. ASR

Acoustic training data for our ASR sytems were harvested from 1787 TED talks. We applied the same alignment and closed caption filtering process as we did in IWSLT 2013 [21], yielding 336 hours of audio.

An i-vector system was first developed on the TED data using Hidden Markov Model Toolkit(HTK)[4] Mel-Frequency Cepstral Coeffiect (MFCC) features and the MIT-LL i-vector software. The elements of the 50 dimensional MFCC vector were based on those used by the ALIZE toolkit [22]. Non-speech frames were removed using the word alignments from the closed caption filtering process, and the features were normalized to zero mean and unit variance on a per-speaker basis. The universal background model included 1024 Gaussians with diagonal covariances, and the i-vector dimension was set to 100. Lastly, the Eigen Factor Radial method [22] was applied to normalize the i-vectors.

A hybrid deep neural-net(DNN) - hidden Markov model(HMM) speech recognition system was developed using Theano and a version of HTK that we modified according to the method of [23]. A context window of 9 frames was used on the input, and the speaker-specific i-vector was appended to each set of stacked features [24]. The feature set consisted of 24 log filterbank outputs with delta and acceleration coefficients; the features were normalized to zero mean and unit variance on a per-speaker basis. The DNN included 5 hidden layers with 1024 rectified linear units per hidden layer and 8000 output units. The network weights were initialized as suggested in [25]. Cross-entropy training was performed using a minibatch size of 512 and an initial learning rate of 0.0005 that was adjusted according to the QuickNet newbob algorithm.[5]

LM data selection was implemented using the same procedure as our IWSLT 2014 system. Interpolated trigram and 4-gram LMs were estimated on TED, 1/8 of Gigaword, and 1/8 of News 2007–2014 using the SRILM Toolkit. A RNN maximum entropy LM was estimated on the same set of training texts using the RNNLM Toolkit. 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 100,000 words.

Automatic segmentation of the test data was performed using the same procedure as in IWSLT 2014[3], except that we padded the speech end points by 0.25 seconds (instead of 0.15 seconds). Recognition lattices were produced using HDecode with the trigram LM and then rescored with the 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 probabilites from the 4-gram and RNN LM. Interpolation weights of 0.25 for the 4-gram and 0.75 for the RNN were chosen based on results from previous experiments.

Adaptation data was selected for each speaker using confidence scores [26]. In our work, we estimated confidence scores at the acoustic frame level by aligning the 20-best hypotheses for each utterance and counting the number of matching HMM shared states. Next, speaker-dependent DNNs were estimated on frames with a confidence score of 0.9 or higher. For each speaker, the initial DNN was updated using a learning rate of 0.0000625 and a single epoch of training. The test set was then decoded a second time and LM rescoring was reapplied.

A second ASR system was built using the Kaldi open source speech recognition toolkit [27]. This system was based on the LIUM recipe as released with Kaldi under egs/tedlium/s5. The details of the particular system used for the IWSLT 2015 Kaldi-based ASR system are as follows. The acoustic model training data and LM data matched exactly what was used as previously described in the HTK ASR system. The first step was to build a network to produce bottleneck (BN) features [28]. MFCCs from 40 filterbanks and 3 pitch features were used as input to a neural network of 2 hidden layers each of dimension 1500 with a 40 dimension BN layer producing the output features. These 40 BN features were then used to build a GMM-HMM. Speaker adaptive training was then conducted on this GMM-HMM using feature-space maximum likelihood linear regression (fMLLR) transforms. These models were then used to train a DNN of the Deep Belief Network (DBN)

[4] http://htk.eng.cam.ac.uk

[5] http://www.icsi.berkeley.edu/Speech/faq/nn-train.html

ASR System	Decode	4-gram	4-gram+RNN
HTK first-pass	13.7	13.0	11.9
HTK	11.3	10.9	10.0
Kaldi	13.3	12.6	11.4

Table 8: English tst2013 WER.

variety described as having 6 hidden layers with 2048 neurons per layer. Four additional iterations using the state-level Minimum Bayes Risk (sMBR) discriminative crierion were then executed. This system was evaluated using the trigram LM to produce recognition lattices, which were then rescored with the 4-gram and RNN LMs as described for the HTK system.

Table 8 shows the WER of each system on tst2013 after evaluating the decoder, rescoring with the 4-gram LM, and interpolating the 4-gram and RNN LM scores. For comparison purposes, we included the results of the HTK system prior to updating the weights of the DNN (denoted as HTK first-pass). The final hypothesis was selected by applying N-best ROVER to the output from the HTK system and the Kaldi system. This yielded a 9.4% WER on tst2013 and a 6.6% WER on tst2015.

4. SLT

New for this year, we combine our efforts in ASR and MT to produce an entry to the SLT task for the English-Chinese language pair. We use the rover output from system combination from the ASR task and translate it with a variant of our best English-Chinese MT system. For segmenting the output, we used the segmentations produced by the ASR system, based on lengths of pauses. To repunctuate the ASR output, we created a classifier based on a recurrent neural network. For each word, the classifier reports which punctuation, if any, follows it. The output layer is a softmax over a limit set (period, comma, question mark, exclamation point, and no punctuation). The inputs to the classifier are a gated recurrent unit [29] hidden state for the word in question, as well as its word vector and the word vectors for the following two words after. Our experimentation was quite limited, but we observed that (a) having the word vector as well as the recurrent state for the current word was helpful, and (b) three layer deep gated recurrent unit worked best out of 1-4. The system was trained on the English side of TED data, with 600-dimensional word vector size, and vocabulary of about 60K words. We did not try any other repunctuation techniques.

One of our alternate approaches for adapting ASR output to MT involves taking the output of the ASR system when decoding dev2010 then using the mwerAlign[30] program to then fit the ASR output segments to the English portion of the dev2010 tuning set. We then tune the MT system with this new dev set in order to better match the ASR English output to the English–Chinese MT system. We submitted this as a contrastive system.

5. Future Research

For future research, we are beginning to look at the metadata for the individual TED talks. We have looked at the distribution of dev, test and training talks by date posted, for example. We are particularly interested in the way different translators may affect the quality of the translations. The TED website assigns a translator ID to each translator, which can be used to isolate his or her talks. The IWSLT files provide the translator metadata for the training files; for the dev and test files, it was necessary to look up the translator annotations in the source files on the TED Talks website.

We looked at the output for each talk in the test files individually, and compared scores for different translators. For example, the scores in cased BLEU for tst2011 are shown in Tables 9 and 10 respectively.

BLEU	translator ID	Set of Talk ID's
13.62	221131	1137, 1176, 1160, 1165
16.33	495543	1104, 1115, 1107
16.72	220760	1102, 1171

Table 9: tst2011 scores for multiple talks by a single translator measured in BLEU.

The individual scores cover a surprising range; one question we want to explore is whether this reflects difficulty in the topic, expertise of the translator, or some combination of these. In Table 10, we see that a single translator can have a wide range of BLEU scores over different talks.

Next, we looked at the distribution of these translators in the training data. For the translators who did at least two talks in the test sets (tst2010 through tst2014), we found that some had translated only a few of the training documents, while others had translated twenty or more documents as shown in Table 11.

While there is not enough training data by translator to train an entire system, there is enough data to try to create an MT system that is tuned to a particular translator. We compared a system adapted from System 0 from Table 4 with a variant system in which we held out training files from a particular translator to use as a dev set.

This creates four possibilities, as shown in Table 12. We can train on the original training files, or hold out the training documents by the specified translator; we can tune on dev2010, or on the held out training documents. We tested this approach using translator 495543 and translator 354776. Translator 495543 had three talks in tst2011 that scored well in the original system; translator 364776 had 1 talk in tst2011 that

BLEU	talkid	URL	translator ID
20.65	1104	eythor_bender_demos_human_exoskeletons	495543
9.95	1096	mark_bezos_a_life_lesson_from_a_volunteer_firefighter	193561
16.66	1102	isabel_behncke_evolution_s_gift_of_play_from...	220760
12.36	1166	alice_dreger_is_anatomy_destiny	831361
11.21	1161	jessi_arrington_wearing_nothing_new	925579
11.64	1137	carlo_ratti_architecture_that_senses_and_responds	221131
15.57	1171	camille_seaman_haunting_photos_of_ice	220760
17.01	1115	mick_ebeling_the_invention_that_unlocked_a_locked...	495543
17.24	1176	jok_church_a_circle_of_caring	221131
13.53	1107	ralph_langner_cracking_stuxnet_a_21st_century...	495543
10.02	1114	morgan_spurlock_the_greatest_ted_talk_ever_sold	354776
15.44	1144	amit_sood_building_a_museum_of_museums_on_the...	250727
16.80	1160	aaron_o_connell_making_sense_of_a_visible_quantum...	221131
12.15	1165	paul_romer_the_world_s_first_charter_city	221131

Table 10: `tst2011` per-talk scores measured in cased BLEU.

	Test		Train	
translator ID	docs	lines	docs	lines
221131	4	344	12	1346
1077318	2	330	4	429
495543	3	235	29	3248
250727	3	192	34	3995
1636197	2	167	26	2232
1648682	2	167	7	1022
1213653	2	147	21	2046
1053094	2	122	14	1806
220760	2	84	41	4373

Table 11: Distribution of Translator effort across test and train sets.

docs 1104, 1115, 1107;	**translator=495543**
dev2010	dev=495543
train (all) 15.89	15.29
train – 495543 13.29	14.08
docs 1114;	**translator=354776**
dev2010	dev=354776
train (all) 9.89	8.64
train – 354776 8.77	8.99

Table 12: Effect of translator-specific tuning on scores for specified `tst2011` documents reported in cased BLEU.

	dev2010	dev=495543	dev=354776
train (all)	16.70	13.39	11.25
train – 495543	13.94	14.90	–
train – 354776	14.84	–	15.10

Table 13: Effect of translator-specific tuning on scores on full `tst2011` reported in cased BLEU.

scored poorly in the original system. The held-out data for translator 495433 had 3248 lines; the held-out data for translator 354776 had 5335 lines.

When training on the restricted training data, we see an improvement for both translators in tuning on the held out data instead of `dev2010`. However, this tuning improvement is not enough to offset an overall drop in score from the reduction in training data.

Looking at scores for the complete `tst2011` test set, shown in Table 13, we see an expected drop in BLEU when restricting the training data (Column 1) and we continue to see an improvement in score when tuning on one of the `tst2011` translators instead of tuning with `dev2010` (Rows 2 and 3). Similar improvements with translator-specific tuning were seen for `dev2010`, `tst2012`, and `tst2013`, even though those test sets do not contain talks by these particular translators.

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 26 Oct 2015. Originator Reference Number: RH-15-114705. Case Number: 88ABW-2015-5214.

6. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2015 Evaluation Campaign," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT'15)*, ser. Proceedings of IWSLT, 2015.

[2] M. Cettolo, C. Girardi, and M. Federico, "WIT3:

Web Inventory of Transcribed and Translated Talks," ser. Proceedings of EAMT, 2012, pp. 261–268.

[3] M. Kazi, E. Salesky, B. Thompson, J. Ray, M. Coury, T. Shen, Wade Anderson, G. Erdmann, J. Gwinnup, K. Young, B. Ore, and M. Hutt, "The MIT-LL/AFRL IWSLT-2014 MT system," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT'14)*, Lake Tahoe, California, December 2014.

[4] P.-C. Chang, M. Galley, and D. C. Manning, *Proceedings of the Third Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2008, ch. Optimizing Chinese Word Segmentation for Machine Translation Performance, pp. 224–232.

[5] W. A. Gale, "Good-turing smoothing without tears," *Journal of Quantitative Linguistics*, vol. 2, 1995.

[6] J. Gwinnup, T. Anderson, G. Erdmann, K. Young, C. May, M. Kazi, E. Salesky, and B. Thompson, "The AFRL-MITLL WMT15 system: There's more than one way to decode it!" in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 112–119. [Online]. Available: http://aclweb.org/anthology/W15-3011

[7] 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.

[8] 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.

[9] K. Heafield, P. Koehn, and A. Lavie, "Grouping language model boundary words to speed k-best extraction from hypergraphs," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, USA, June 2013, pp. 958–968.

[10] 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, 2008, pp. 848–856.

[11] M. Junczys-Dowmunt, "Phrasal rank-encoding: Exploiting phrase redundancy and translational relations for phrase table compression," *The Prague Bulletin of Mathematical Linguistics*, vol. 98, pp. 63–74, 2012.

[12] G. Erdmann and J. Gwinnup, "Drem: The AFRL submission to the WMT15 tuning task," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 422–427. [Online]. Available: http://aclweb.org/anthology/W15-3054

[13] 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.

[14] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.

[15] I. Sutskever, O. Vinyals, and Q. V. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[16] 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.

[17] J. Devlin, C. Quirk, and A. Menezes, "Precomputable multi-layer neural network language models," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 256–260. [Online]. Available: http://aclweb.org/anthology/D15-1029

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[19] 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.

[20] A.-V. I. Rosti, B. Zhang, S. Matsoukas, and R. Schwartz, "Incremental hypothesis alignment for building confusion networks with application to machine translation system combination," in *Proceedings of the Third Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2008, pp. 183–186.

[21] 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.

[22] A. Larcher, J.-F. Bonastre, B. Fauve, K. A. Lee, C. Levy, H. Li, J. Mason, and J.-Y. Parfait, "ALIZE 3.0 - Open Source Toolkit for State-of-the-Art Speaker Recognition," in *Proceedings of Interspeech*, Lyon, France, August 2013.

[23] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 30–42, January 2012.

[24] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker Adaptation of Neural Network Acoustic Models using I-Vectors," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomouc, Czech Republic, December 2013.

[25] S. Zhang, H. Jiang, S. Wei, and L.-R. Dai, "Rectified Linear Neural Networks with Tied-Scalar Regularization for LVCSR," in *Proceedings of Interspeech*, Dresden, Germany, September 2015.

[26] P. Shen, X. Lu, X. Hu, N. Kanda, M. Saiko, and Chiori, "The NICT ASR system for IWSLT 2014," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT'14)*, Lake Tahoe, California, December 2014.

[27] 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, December 2011.

[28] F. Grezl and P. Fousek, "Optimizing Bottle-Neck Features for LVCSR," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008.

[29] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[30] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating machine translation output with automatic sentence segmentation," in *International Workshop on Spoken Language Translation*, Pittsburgh, PA, USA, Oct. 2005, pp. 148–154.

The Edinburgh Machine Translation Systems for IWSLT 2015

Matthias Huck, Alexandra Birch

School of Informatics
University of Edinburgh
Scotland, United Kingdom
mhuck@inf.ed.ac.uk a.birch@ed.ac.uk

Abstract

This paper describes the University of Edinburgh's machine translation (MT) systems for the IWSLT 2015 evaluation campaign. Our submissions are based on preliminary systems which are under development for the purpose of lecture translation in the TraMOOC project,[1] funded by the European Union.

We participated in the English→Chinese and the English→German translation tasks in the MT track, utilizing only data supplied by the organizers or listed as permissible. We built phrase-based translation systems for both tasks. For English→German, we furthermore made use of 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, 3] for the English→Chinese and English→German translation tasks, and additionally a string-to-tree syntax-based system [4, 5] for English→German. Our primary submission translations for English→Chinese are the output of a single phrase-based system, whereas our primary submission translations for English→German are the output of a system combination [6] of two phrase-based systems and one syntax-based system.

The setups for our phrase-based systems have evolved from the configurations of the engines we built for Edinburgh's participation in last year's IWSLT evaluation [7] and in this year's Workshop on Statistical Machine Translation (WMT) shared translation task [8].

Edinburgh's syntax-based systems have recently yielded state-of-the-art performance on English→German news translation tasks [9, 10] and have been applied in an IWSLT-style setting for the first time for our last year's contrastive submission [7]. This year, a syntax-based system became part of our primary submission by contributing input to a system combination.

For system combination, we employed the implementation that has been released as part of the Jane machine translation toolkit [11]. Multiple previous top-ranked submissions to open evaluation campaigns have relied on this system combination framework [12, 13, 14].

[1] http://tramooc.eu

2. System Overview

2.1. Training and Tuning

For both the phrase-based systems and the syntax-based system, we first preprocess the parallel training data and then create word alignments by aligning the data in both directions with MGIZA++ [15]. We use a sequence of IBM word alignment models [16] with five iterations of EM training [17] of Model 1, three iterations of Model 3, and three iterations of Model 4. After EM, we obtain a symmetrized alignment by applying the `grow-diag-final-and` heuristic [18, 3] to the two trained alignments. We extract bilingual phrases that are consistent with the symmetrized word alignment from the parallel training data. In the case of the syntax-based system, we also need syntactic parses of the target-language side of the parallel training data in order to extract synchronous context-free grammar rules.

We train n-gram language models (LMs) with modified Kneser-Ney smoothing [19, 20]. KenLM [21] is employed for LM training and scoring, and SRILM [22] for linear LM interpolation.

Our translation model incorporates a number of different features in a log-linear combination [23]. We tune the feature weights with batch k-best MIRA [24] to maximize BLEU [25] on a development set. We run MIRA for 25 iterations on 200-best lists (phrase-based) or 1000-best lists (syntax-based).

In our experiments (cf. Section 3) with the phrase-based system, we commence with a plain baseline which comprises a small amount of vital features only. We then incrementally extend the system with further features and more advanced techniques. Each setup is re-tuned individually to obtain optimal feature weights for the respective configuration.

Proceedings of the 12th International Workshop on Spoken Language Translation
Da Nang, Vietnam, December 3-4, 2015

2.2. Phrase-based System

The features of our plain phrase-based baseline are:

- Phrase translation log-probabilities in both target-to-source and source-to-target direction.
- Lexical translation log-probabilities in both target-to-source and source-to-target direction.
- Word penalty.
- Phrase penalty.
- A distance-based distortion cost.
- A 5-gram language model over words. Singleton *n*-grams of order three and higher are discarded.

We extract phrases up to a length of five. We prune the phrase table to a maximum of 100 best translation options per distinct source side and apply a minimum score threshold τ on the source-to-target phrase translation probability, with $\tau = 0.0001$ during tuning and $\tau = 0.00001$ during testing. We use cube pruning [26] in decoding. Pop limit and stack limit are set to 1000 for tuning and to 5000 for testing. A distortion limit of six is enforced during decoding, and we disallow reordering over punctuation. Furthermore, Minimum Bayes Risk decoding [27] is employed for testing.

Extensions we experimented with for either English→German or English→Chinese are:

LRM. A hierarchical lexicalized reordering model [28]. This model estimates the probabilities of orientation classes for each phrase from the training data. We use four orientation classes: *monotone*, *swap*, *left-discontinuous*, and *right-discontinuous*.

TM factors. Translation model (TM) factors beyond word surface forms [29, 30]. Factors can for instance be part-of-speech (POS) tag, morphological tag, or automatically learnt word classes, e.g. from `mkcls` [31]. Factors can be added on either source side or target side or both. We do not use a generation step but merely enrich the phrases with factored annotation. The annotation is obtained by tagging the training data prior to phrase extraction. Source-side factors such as POS or morphological tags can be helpful for disambiguating phrases: at decoding time, we annotate the input text in a preprocessing step, and the decoder only applies phrases with matching annotation. Target-side factors can be helpful for providing a longer context window via *n*-gram models of higher order over representations given by the factors (which we mention next in this list).

7-gram class-based LM. A 7-gram language model over `mkcls` word classes.

7-gram POS LM. A 7-gram language model over part-of-speech tags.

7-gram morph LM. A 7-gram language model over morphological tags.

Good-Turing smoothing. Good-Turing smoothing of phrase translation probabilities [32].

Count features. Seven binary features indicating absolute occurrence count classes of phrase pairs.

Sparse features. Sparse phrase length features, and sparse lexical features for the top 200 words.

Domain indicators. Binary features indicating the provenance of phrase pairs: if a phrase pair has been seen in a particular training corpus, a binary indicator associated with the respective training corpus fires on application of that phrase pair during decoding.

Phrase table fill-up. A foreground phrase table extracted from in-domain data is filled up with entries from a background phrase table extracted from all data [33, 34]. An entry from the background table is only added if the foreground table does not know the respective phrase identity. A binary feature distinguishes background phrases from foreground phrases. (The baseline uses a phrase table extracted from all data.)

5-gram OSM. A 5-gram operation sequence model [35].

5-gram OSM over word classes. A 5-gram operation sequence model over `mkcls` word classes.

5-gram OSMs over factors. Operation sequence models over various representations given by the factors.

In-domain OSMs. 5-gram operation sequence models over words and factors, trained on the in-domain portion of the parallel data only.

Unpruned LM. The baseline 5-gram language model over words is replaced by a version where singleton *n*-grams of order three and higher have not been discarded.

No singleton phrases. Phrase pairs with an occurrence count of one are removed from the phrase table.

Sparse LR. Sparse lexicalized reordering features [36] with weights learnt via RPROP with a maximum expected BLEU objective [37, 38]. The features are added on top of the standard hierarchical lexicalized reordering model. We apply features based on all words as well as word classes with 200 clusters on both source and target side. Active feature groups are *between*, *phrase*, and *stack*. We follow a similar training procedure as suggested by Wuebker et al. [38].[2] Maximum expected BLEU training with RPROP is conducted on the in-domain fraction of the training data. We train on 100-best lists. We set the regularization parameter to 10^{-5} and use the weights obtained after 50 iterations of RPROP. Rather than decoding the training data with leaving-one-out, we utilize a system with no singleton phrases. The learnt sparse lexicalized reordering features are condensed to a single feature per orientation, as suggested by Auli et al. [37]. A final MIRA run tunes weights for those condensed features along with the other features in the log-linear model of the translation system.

[2] Our tool for maximum expected BLEU training has been released as part of the Moses code base on GitHub.

2.3. Syntax-based System

The syntactic translation model for our string-to-tree system conforms to the GHKM syntax approach as proposed by **G**alley, **H**opkins, **K**night, and **M**arcu [4] with composed rules [39, 40]. Decoding is carried out with a procedure based on bottom-up chart parsing. The parsing algorithm is extended to handle translation candidates and to incorporate language model scores via cube pruning [26].

Standard features of Edinburgh's string-to-tree syntax-based systems are:

- Rule translation log-probabilities in both target-to-source and source-to-target direction, smoothed with Good-Turing discounting.
- Lexical translation log-probabilities in both target-to-source and source-to-target direction.
- Word penalty.
- Rule penalty.
- A rule rareness penalty.
- The monolingual PCFG probability of the tree fragment from which the rule was extracted.
- A 5-gram language model over words.

When extracting syntactic rules, we impose several restrictions for composed rules, in particular a maximum number of 100 tree nodes per rule, a maximum depth of seven, and a maximum size of seven. We discard rules with non-terminals on their right-hand side if they are singletons in the training data. Only the 200 best translation options per distinct rule source side with respect to the weighted rule-level model scores are loaded by the decoder. Search is carried out with a maximum chart span of 25, a rule limit of 500, a stack limit of 200, and a pop limit of 1000 for cube pruning [41]. During tuning, we constrain the translation options per rule source side to the top 20 candidates for faster optimization, and we set the cube pruning pop limit to 500. We configure Moses' `n-best-factor` parameter at a value of 100 to avoid short *n*-best lists.

For our IWSLT English→German syntax-based system, the target side of the parallel training data is parsed with BitPar [42]. We remove grammatical case and function information from the annotation obtained with BitPar and apply right binarization of the German parse trees prior to rule extraction [43, 44, 45].

The system is adapted to the TED domain by extracting two separate rule tables (from in-domain data and from out-of-domain parallel data) and merging them with a fill-up technique [33]. We also integrate a second 5-gram LM trained on the in-domain corpus into the log-linear combination. Additionally we add soft source syntactic constraints [46] and augment the system with non-syntactic phrases [47].

2.4. System Combination

The Jane machine translation toolkit implements a system combination approach via confusion network decoding [11]. The hypotheses from individual MT systems are aligned to each other with METEOR [48]. A confusion network is generated which represents all combined translations that can be produced from the set of individual hypotheses. The optimal combined hypothesis is chosen by finding the best path through the confusion network. The decision process is guided by a couple of simple features:

- Binary system voting features.
- A primary system indicator.
- Word penalty.
- A small 3-gram language model trained only on the set of individual hypotheses.
- A conventional 5-gram language model.

Feature weights are optimized with MERT [49].

We combine three individual systems with this method for our English→German primary submission.

3. Experiments

3.1. English→German MT

For the English→German MT task, we submitted outputs of two different phrase-based systems (*contrastive 1* and *contrastive 2*), a syntax-based system (*contrastive 3*), and a system combination (*primary*) of those three single systems. Table 1 shows their respective performance in terms of BLEU scores, along with the official scores [50] of the best last year's submission for comparison.

Our English→German systems are trained using monolingual and parallel data from the in-domain WIT[3] corpus [52], as well as Europarl [53], MultiUN [54], the parallel corpus from the Wikipedia [55] as provided for the evaluation campaign, the German Political Speeches corpus [56], and the permissible corpora from the WMT shared translation task [57]. For the systems with factors, annotation exploited in addition to word surface forms is: part-of-speech tags [58] on the English side; morphological tags [59] and part-of-speech tags [59] on the German side; and word classes from `mkcls` with 50 clusters on both sides.

5-gram LMs over words are estimated over a concatenation of all target-language training data, rather than linearly interpolating individual LMs over the different corpora. We found this to perform equally well or better on the given task. Class-based LMs, POS LMs, and morph LMs, on the other hand, are linear interpolations of individually trained LMs.[3] Feature weights for all single engines are tuned on a concatenation of `TED.dev2010`, `TED.tst2010`, `TEDX.dev2012`, and `TEDX.tst2013`.[4]

[3] Individual LMs over factors are trained with KenLM's `--discount_fallback --prune '0 0 1'` parameters.

[4] Note that `TEDX.tst2013` and `tst2013` (= `TED.tst2013`) are two different sets.

en→de	tst2011	tst2012	tst2013	tst2014	tst2015
phrase-based (*contrastive 1*)	28.3	24.7	26.3	23.3	25.4
phrase-based w/o singleton phrases + sparse LR (*contrastive 2*)	27.9	24.5	26.8	23.3	25.5
syntax-based (*contrastive 3*)	26.8	23.6	26.1	22.7	24.3
system combination (*primary*)	28.4	25.6	27.0	24.0	26.0
best IWSLT 2014 submission (*EU-BRIDGE* [14])	–	–	26.2	23.3	–

Table 1: Edinburgh submission system results for the English→German MT task (case-sensitive BLEU scores), and results of the best IWSLT 2014 submission as reported by Cettolo et al. [50]. The Edinburgh *primary* submission is a system combination of the three *contrastive* systems and was tuned on tst2012.

en→zh	tst2012	tst2013	tst2014	tst2015
phrase-based (*primary*)	21.3	22.9	19.6	25.4
best IWSLT 2014 submission (*USTC* [51])	–	22.5	21.6	–

Table 2: Edinburgh submission system results for the English→Chinese MT task (character-based BLEU scores), and results of the best IWSLT 2014 submission as reported by Cettolo et al. [50].

Phrase-based system. Table 3 presents the results achieved with the plain phrase-based baseline, and the gains when incrementally adding extensions as described in Section 2.2.[5] The *contrastive 1* submission system outperforms the plain baseline by up to +3.6 BLEU points (on `tst2011`). If we remove singleton phrases on top of that, we observe a small gain on `tst2013`, but performance degrades slightly on `tst2011` and `tst2012`. The sparse lexicalized reordering features trained via RPROP with a maximum expected BLEU objective (*contrastive 2*) do not further affect the results too much.[6] However, the *contrastive 2* submission system outperforms the plain baseline by +3.5 BLEU points on a different test set (on `tst2013`).

Syntax-based system. In the syntax-based system, we utilize neither the parallel corpus from the Wikipedia nor MultiUN or the German Political Speeches corpus for rule extraction.[7] We only use the target side of the Wikipedia corpus as LM training data. The development set is the same as for the phrase-based systems. Our IWSLT string-to-tree syntax-based system (*contrastive 3*) is outperformed by the phrase-based submission systems by a bit more than one BLEU point on this year's evaluation set (`tst2015`), cf. Table 1. The average BLEU delta on the other test sets is lower, though.

System combination. The parameters of the system combination (*primary*) are optimized on `tst2012`. The consensus translation produced by the system combination boosts the BLEU score by half a point over the best single system on this year's evaluation set (`tst2015`), cf. Table 1. Improvements on the other test sets vary between +0.1 and

[5]The order in which extensions are added is not motivated by any specific rationale other than our personal preference.

[6]We add the *sparse LR* to the system without singleton phrases. This avoids a mismatch with the system used in *n*-best generation for maximum expected BLEU training.

[7]Due to time constraints, these corpora have been omitted for the benefit of faster training.

en→de	tst2011	tst2012	tst2013
phrase-based baseline	24.7	22.0	23.3
+ LRM	25.5	22.0	24.1
+ TM factors	25.3	22.1	23.8
+ 7-gram class-based LM	25.9	22.5	24.2
+ 7-gram POS LM	26.1	22.8	24.6
+ 7-gram morph LM	26.5	22.9	24.9
+ Good-Turing smoothing	26.8	23.6	24.9
+ count features	26.8	23.4	24.9
+ sparse features	26.9	23.7	25.1
+ domain indicators	27.2	23.6	25.3
+ 5-gram OSM	27.6	24.1	26.1
+ 5-gram OSMs over factors	27.8	24.3	26.0
+ in-domain OSMs	28.0	24.3	26.3
+ unpruned LM (*contrastive 1*)	28.3	24.7	26.3
+ no singleton phrases	27.9	24.6	26.7
+ sparse LR (*contrastive 2*)	27.9	24.5	26.8

Table 3: Incremental improvements over a plain phrase-based baseline for English→German (case-sensitive BLEU scores).

en→zh	tst2012	tst2013
phrase-based baseline	19.2	21.0
+ LRM	19.8	21.7
+ Good-Turing smoothing	20.0	21.9
+ count features	20.1	21.9
+ 7-gram class-based LM (in-domain)	20.0	22.0
+ phrase table fill-up	21.0	22.3
+ 5-gram OSM	21.0	22.5
+ 5-gram OSM over word classes	20.9	22.5
+ in-domain OSMs (*primary*)	21.3	22.9

Table 4: Incremental improvements over a plain phrase-based baseline for English→Chinese (character-based BLEU scores).

+0.7 (disregarding tst2012, since it has been used to tune the system combination).

Our best single system yields translation quality on the level of the last year's best submission, which was a system combination [14]. Our primary submission is around 0.7 BLEU points better than last year's best submission.

3.2. English→Chinese MT

For the English→Chinese MT task, we submitted the output of a phrase-based single system (*primary*). Table 2 shows the performance in terms of BLEU scores, measured on character level with the aid of the Chinese character tokenization script provided by the organizers of the evaluation campaign. For comparison, we also include the official scores [50] of the best last year's submission.

Our English→Chinese systems are trained using monolingual and parallel data from the in-domain WIT^3 corpus [52], as well as MultiUN [54]. For the English-Chinese MultiUN parallel data, we resorted to the sentence-aligned version as distributed in OPUS [60]. We perform Chinese word segmentation with the Stanford Word Segmenter [61] as a preprocessing step on all target-side data. The character-based tokenization is conducted for evaluation purposes only, whereas our models operate on word-segmented data.

Table 4 presents the results achieved with the plain phrase-based baseline, and the gains when incrementally adding extensions as described in Section 2.2. The 5-gram LM over words is a linear interpolation of individual LMs, the 7-gram class-based LM is trained on in-domain data only. The only factors we use for English→Chinese are word classes from mkcls with 50 clusters. Feature weights are tuned on a concatenation of dev2010, tst2010, and tst2011. The submission system outperforms the plain baseline by up to +2.1 BLEU points (on tst2012).

The comparison with last year's best submission [51] is somewhat surprising: the BLEU score of our system is +0.4 points higher on tst2013, but we significantly lag behind on tst2014. We are currently unaware of the reason for this behavior.

4. Summary

We built high-quality machine translation systems for the IWSLT 2015 English→Chinese and English→German translation tasks in the MT track. By utilizing advanced features and techniques, we have been able to achieve improvements over plain phrase-based baselines of two BLEU points or more on both language pairs. All methods we employed are implemented in publicly available software such as the Moses and the Jane statistical machine translation toolkits.

5. Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 644333 (*TraMOOC*).

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. 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.

[2] R. Zens, F. J. Och, and H. Ney, "Phrase-Based Statistical Machine Translation," in *German Conf. on Artificial Intelligence*, Aachen, Germany, Sept. 2002, pp. 18–32.

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

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

[5] P. Williams and P. Koehn, "GHKM Rule Extraction and Scope-3 Parsing in Moses," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Montréal, Canada, June 2012, pp. 388–394.

[6] E. Matusov, G. Leusch, R. E. Banchs, N. Bertoldi, D. Dechelotte, M. Federico, M. Kolss, Y.-S. Lee, J. B. Marino, M. Paulik, S. Roukos, H. Schwenk, and H. Ney, "System Combination for Machine Translation of Spoken and Written Language," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 7, pp. 1222–1237, Sept. 2008.

[7] 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)*, Lake Tahoe, CA, USA, Dec. 2014, pp. 49–56.

[8] B. Haddow, M. Huck, A. Birch, N. Bogoychev, and P. Koehn, "The Edinburgh/JHU Phrase-based Machine Translation Systems for WMT 2015," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Lisbon, Portugal, Sept. 2015, pp. 126–133.

[9] 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.

[10] P. Williams, R. Sennrich, M. Nadejde, M. Huck, and P. Koehn, "Edinburgh's Syntax-Based Systems at WMT 2015," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Lisbon, Portugal, Sept. 2015, pp. 199–209.

[11] M. Freitag, M. Huck, and H. Ney, "Jane: Open Source Machine Translation System Combination," in *Proc. of the Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, Gothenburg, Sweden, Apr. 2014, pp. 29–32.

[12] 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.

[13] 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.

[14] 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)*, Lake Tahoe, CA, USA, Dec. 2014, pp. 57–64.

[15] 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.

[16] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, June 1993.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statist. Soc. Ser. B*, vol. 39, no. 1, pp. 1–22, 1977.

[18] 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.

[19] R. Kneser and H. Ney, "Improved Backing-Off for M-gram Language Modeling," in *Proceedings of the Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, Detroit, MI, USA, May 1995, pp. 181–184.

[20] 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.

[21] K. Heafield, "KenLM: Faster and Smaller Language Model Queries," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Edinburgh, Scotland, UK, July 2011, pp. 187–197.

[22] A. Stolcke, "SRILM – an Extensible Language Modeling Toolkit," in *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP)*, vol. 3, Denver, CO, USA, Sept. 2002.

[23] 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.

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

[25] 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.

[26] D. Chiang, "Hierarchical Phrase-Based Translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, June 2007.

[27] S. Kumar and W. Byrne, "Minimum Bayes-Risk Decoding for Statistical Machine Translation," in *Proc. of the Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Boston, MA, USA, May 2004, pp. 169–176.

[28] 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.

[29] P. Koehn and H. Hoang, "Factored Translation Models," in *Proc. of the 2007 Joint Conf. on Empirical Methods in Natural Language Processing and*

Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, June 2007, pp. 868–876.

[30] 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.

[31] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes," in *Proc. of the Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, Bergen, Norway, June 1999, pp. 71–76.

[32] G. Foster, R. Kuhn, and H. Johnson, "Phrasetable Smoothing for Statistical Machine Translation," in *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, Sydney, Australia, July 2006, pp. 53–61.

[33] 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.

[34] 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)*, San Diego, CA, USA, Oct./Nov. 2012.

[35] N. Durrani, A. Fraser, and H. Schmid, "Model With Minimal Translation Units, But Decode With Phrases," in *Proc. of the Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, GA, USA, June 2013, pp. 1–11.

[36] C. Cherry, "Improved Reordering for Phrase-Based Translation using Sparse Features," in *Proc. of the Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, GA, USA, June 2013, pp. 22–31.

[37] 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, pp. 1250–1260.

[38] J. Wuebker, S. Muehr, P. Lehnen, S. Peitz, and H. Ney, "A Comparison of Update Strategies for Large-Scale Maximum Expected BLEU Training," in *Proc. of the Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Denver, CO, USA, May 2015, pp. 1516–1526.

[39] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer, "Scalable Inference and Training of Context-Rich Syntactic Translation Models," in *Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the Assoc. for Computational Linguistics*, Sydney, Australia, July 2006, pp. 961–968.

[40] S. DeNeefe, K. Knight, W. Wang, and D. Marcu, "What Can Syntax-Based MT Learn from Phrase-Based MT?" in *Proc. of the 2007 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, June 2007, pp. 755–763.

[41] M. Huck, D. Vilar, M. Freitag, and H. Ney, "A Performance Study of Cube Pruning for Large-Scale Hierarchical Machine Translation," in *Proceedings of the NAACL 7th Workshop on Syntax, Semantics and Structure in Statistical Translation*, Atlanta, GA, USA, June 2013, pp. 29–38.

[42] 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.

[43] W. Wang, K. Knight, and D. Marcu, "Binarizing Syntax Trees to Improve Syntax-Based Machine Translation Accuracy," in *Proc. of the 2007 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, June 2007, pp. 746–754.

[44] W. Wang, J. May, K. Knight, and D. Marcu, "Restructuring, Re-labeling, and Re-aligning for Syntax-based Machine Translation," *Computational Linguistics*, vol. 36, no. 2, pp. 247–277, June 2010.

[45] 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.

[46] M. Huck, H. Hoang, and P. Koehn, "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.

[47] ——, "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.

[48] M. Denkowski and A. Lavie, "Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of

Machine Translation Systems," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Edinburgh, Scotland, UK, July 2011, pp. 85–91.

[49] F. J. Och, "Minimum Error Rate Training for Statistical Machine Translation," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003, pp. 160–167.

[50] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 11th IWSLT Evaluation Campaign, IWSLT 2014," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, USA, Dec. 2014, pp. 2–17.

[51] S. Wang, Y. Wang, J. Li, Y. Cui, and L. Dai, "The USTC Machine Translation System for IWSLT 2014," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, CA, USA, Dec. 2014, pp. 134–138.

[52] 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.

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

[54] A. Eisele and Y. Chen, "MultiUN: A Multilingual Corpus from United Nation Documents," in *Proc. of the Int. Conf. on Language Resources and Evaluation (LREC)*, May 2010, pp. 2868–2872.

[55] K. Wołk and K. Marasek, "Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs," *Procedia Technology*, vol. 18, pp. 126–132, 2014, International workshop on Innovations in Information and Communication Science and Technology (IICST), 3-5 September 2014, Warsaw, Poland.

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

[57] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, M. Huck, C. Hokamp, P. Koehn, V. Logacheva, C. Monz, M. Negri, M. Post, C. Scarton, L. Specia, and M. Turchi, "Findings of the 2015 Workshop on Statistical Machine Translation," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Lisbon, Portugal, September 2015, pp. 1–46.

[58] A. Ratnaparkhi, "A Maximum Entropy Part-Of-Speech Tagger," in *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, Philadelphia, PA, USA, May 1996.

[59] H. Schmid, "LoPar: Design and Implementation," Institute for Computational Linguistics, University of Stuttgart, Bericht des Sonderforschungsbereiches "Sprachtheoretische Grundlagen für die Computerlinguistik" 149, 2000.

[60] J. Tiedemann, "Parallel Data, Tools and Interfaces in OPUS," in *Proc. of the Int. Conf. on Language Resources and Evaluation (LREC)*, Istanbul, Turkey, May 2012, pp. 2214–2218.

[61] P.-C. Chang, M. Galley, and C. Manning, "Optimizing Chinese Word Segmentation for Machine Translation Performance," in *Proc. of the Workshop on Statistical Machine Translation (WMT)*, Columbus, OH, USA, June 2008, pp. 224–232.

The MLLP ASR Systems for IWSLT 2015

Miguel Ángel del-Agua, Adrià Martínez-Villaronga, Santiago Piqueras
Adrià Giménez, Alberto Sanchis, Jorge Civera, Alfons Juan

Machine Learning and Language Processing
Departament de Sistemes Informàtics i Computaciò
Universitat Politècnica de València, Spain
mdelagua@dsic.upv.es

Abstract

This paper describes the Machine Learning and Language Processing (MLLP) ASR systems for the 2015 IWSLT evaluation campaing. The English system is based on the combination of five different subsystems which consist of two types of Neural Networks architectures (Deep feed-forward and Convolutional), two types of activation functions (sigmoid and rectified linear) and two types of input features (fMLLR and FBANK). All subsystems perform an speaker adaptation step based on confidence measures the output of which is then combined with ROVER. This system achieves a Word Error Rate (WER) of 13.3% on the 2015 official IWSLT English test set.

1. Introduction

TED is a global set of conferences around the world carried out by the non-profit organisation *Sapling Foundation*. Its talks cover a wide range of different topics such as science, culture, economics or politics, always keeping in mind the slogan "ideas worth spreading". The speakers are given a maximum of 18 minutes to present their ideas in the most appealing way they can, typically in a storytelling format.

In order to ensure the maximum spread of these talks, turns out to be essential their transcription and translation. Big efforts have been devoted to this task, such as *The Open Translation Project* (OTP), which aims to reach out to the 4.5 billion people on the planet who do not speak English. Nevertheless, the OTP utilises crowd-based subtitling platforms, powered by volunteers to translate and caption the videos, which is still a very time-consuming task.

TED talks conform a very appropriate case study where new technologies can be applied. Particularly from the machine learning community, the International Workshop on Spoken Language Translation (IWSLT) organises a yearly challenge which aims at evaluating the core technologies in spoken language translation: automatic speech recognition (ASR), machine translation (MT) and spoken language translation (SLT). Automatically transcribing this kind of videos is still a challenging task due to the spontaneous nature of the speech; variety in acoustic conditions, the presence of disfluencies, hesitations and different accents states a great challenge even for cutting-edge technology in automatic automatic speech recognition.

This paper describes the English and German ASR systems developed in the MLLP group for the IWSLT 2015 evaluation campaign. Most effort went into the development of the English recognition system which is based on the ROVER combination of five subsystems. Each of those subsystems was based on hybrid Deep Neural Networks Hidden Markov Models (DNN-HMM) [1] with different input features (MFCCs and filter bank), activation functions (sigmoid and rectified linear) as well as various architectures such as Deep Convolutional Neural Networks (CNN). It is worth noting that all of these systems were entirely trained using our own software; the transLectures-UPV toolkit.

The rest of this paper is organised as follows. Section 2 describes the ASR toolkit used for the experiments. In Section 3 the automatic audio segmentation technique is introduced. Section 4 is devoted to the English transcription system. Similarly, in Section 5 the German ASR system is described. Finally, conclusions are given in Section 6.

2. Translectures-UPV Toolkit

The transLectures-UPV toolkit (TLK) is composed by a set of tools that allows the development of an end-to-end speech recognition system. Its application range extends from feature extraction to HMM and DNN training and decoding. Since last state published of the toolkit [2] new state-of-the-art techniques have been added:

- DNN training and decoding hybrid based systems.
- Support to Convolutional NNs.
- Support to Multilingual NNs.
- DNN speaker adaptation techniques such as output-feature discriminant linear regression (oDLR) [3].
- DNN sequence discriminative training based on Maximum Mutual Information (MMI).

3. Audio Segmentation

The audio segmentation step performed by the MLLP group for English and German can be viewed as a simplified case of ASR, in which the system vocabulary is constituted by the power set of segment classes: speech and background noise.

Provided an audio stream $\boldsymbol{x}$, the segmentation problem can be stated from a statistical point of view as the search of a sequence of class labels $\hat{\boldsymbol{c}}$ so that

$$\hat{\boldsymbol{c}} = \underset{\boldsymbol{c} \in \mathcal{C}^*}{\operatorname{argmax}}\, p(\boldsymbol{x} \mid \boldsymbol{c})\, p(\boldsymbol{c}) \tag{1}$$

where, as in ASR, $p(\boldsymbol{x} \mid \boldsymbol{c})$ and $p(\boldsymbol{c})$ are modeled by acoustic and language models, respectively. In our case, it should be noted that each word is composed by a single phoneme.

Acoustic models were trained on MFCC feature vectors computed from acoustic samples using TLK. We used a 0.97 coefficient pre-emphasis filter and a 25 ms Hamming window that moves every 10 ms over the acoustic signal. From each 10ms frame, a feature vector of 12 MFCC coefficients is obtained using a 26 channel filter bank. Finally, the energy coefficient and the first and second time derivatives of the cepstrum coefficients are added to the feature vector.

Each segment class is represented by a single-state Hidden Markov Model (HMM) without loops, and its emission probability is modeled by a Gaussian Mixture Model (GMM). Acoustic HMM-GMM models were also trained using TLK, which implements the conventional Baum-Welch algorithm.

A 5-gram back-off language model with constant discount was trained on the sequence of class labels using the SRILM toolkit [4]. Finally, the segmentation process (search) was also carried out by the TLK toolkit.

4. English Transcription System

4.1. Acoustic Modeling

In this section the acoustic modeling process for the English system is described. First, the data selected for training is showed as well as the techniques used for its collection. Then, the training procedure is detailed along with all the subsystems associated.

4.1.1. Data Collection

This year, the IWSLT challenge allowed the use of any publicly available data for acoustic modeling, including TED talks without publication date restrictions (except those listed as disallowed). Given these requirements, roughly 400 hours of TED talks were downloaded from its official web-page [5].

The subtitles attached to a large part of the talks neither match the speaker's speech nor the timings. Therefore, a data filtering process is needed, in which those segments with a deficient or non-existent transcription must be removed. This process was performed in a similar manner to the data filtering performed for building the TEDLIUM corpus [6].

First of all, the input audio was segmented and preprocessed according to the caption timings. Secondly, a recognition step was performed using an out-of-domain acoustic model and a finite state language model. This finite state language model was built using the sequence of words from the reference with silence in-between, allowing loops (hesitations), initial state to any word transitions and from any word to final state transitions.

This way, those segments whose recognition does not match the reference suggest that either the timings are wrongly set or the system is unable to recognise the segment due to non-speech audio. Therefore, after decoding, all of these incorrectly recognised segments were removed, which left us a total of 245 hours of clean speech distributed among 1900 talks.

4.1.2. Training

Regarding feature extraction, two types of acoustic features were extracted. The first type of features are Mel-frequency cepstral coefficients (MFCC), which were extracted with a Hamming window of 25 ms, shifted at 10 ms intervals. The MFCC feature consisted of 16 MFCCs and their first and second derivatives (48-dimensional feature vectors). These feature vectors were then normalised by mean and variance at speaker level. After that, a single feature-space Maximum Likelihood Linear Regression (fMLLR) transform for each training speaker was then estimated and applied to perform speaker-adaptive training (SAT). The second type of features are log Mel filter bank (FBANK) with first and second derivatives which left 120 dimension feature vectors.

Five different acoustic models were trained in our system using TLK. All of them consisted of context-dependent Deep Neural Networks (DNNs) following an hybrid approach. To train these models, we first trained a basic context dependent triphone HMM model, after which a second-pass feature-space Maximum Likelihood Linear Regression (fMLLR) was applied. This model yielded a total of 10492 tied states, estimated following a phonetic decision tree approach [7]. It is worth noting that, in order to obtain the best transcription as to better perform fMLLR, an standard DNN was trained using the MFCCs features. The five models were build on top of these HMM acoustic model and followed a three-pass recognition approach as shown in Fig. 1.

From Fig.1, the *fMLLR CD-DNN* module can be switched among the five different acoustic models. Three of them are feed-forward DNNs and the other two are Deep Convolutional Neural Networks (CNNs). From the first set, all models took as input MFCCs feature frames with a window size of 11. Moreover, all three subsystems shared the same topology: $528 - 2048 * 7 - 10492$, i.e., an input layer with 528 neurons, 7 hidden layers with 2048 neurons and an output layer of 10492 neurons. The pre-training phase technique is also shared, which consisted of the Discriminative Pretraining [8] approach. The first system was a DNN with sigmoid activation functions, trained with the cross-entropy

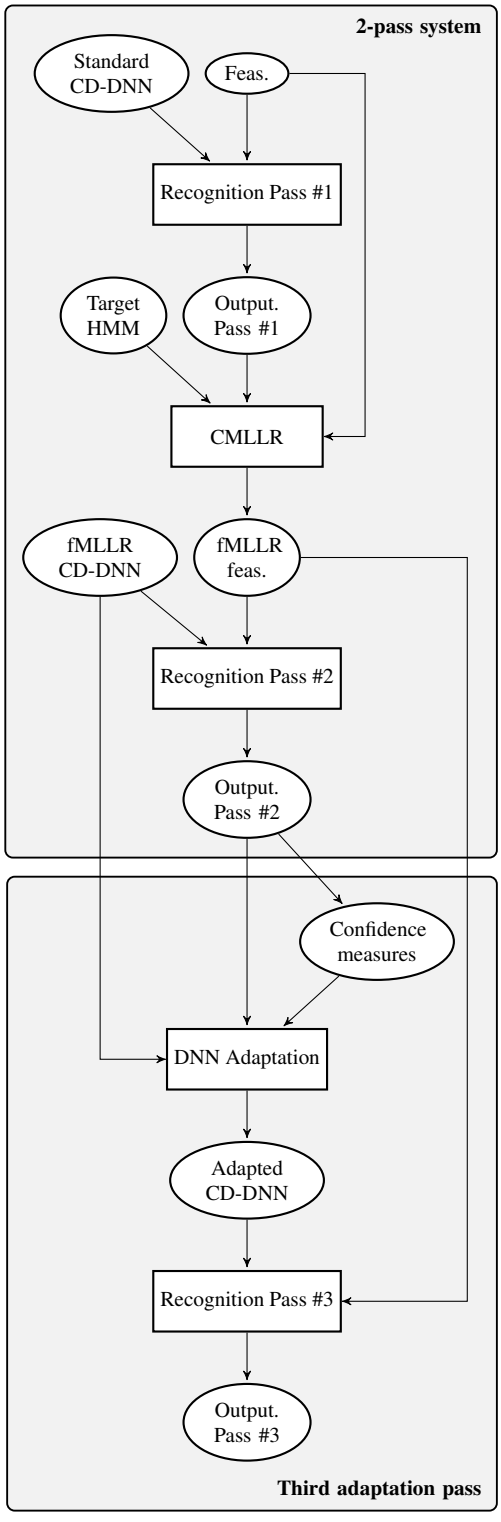


Figure 1: Overview of a multi-pass recognition system including DNN adaptation. Top: 2-pass recognition system using fMLLR features. Bottom: Third pass DNN adaptation.

(CE) criterion (10 epochs) and after that, with sequence discriminative training following the MMI criterion (hereafter DNN-mmi). The second model was a DNN with rectified linear activation functions, trained following the CE criterion during 45 epochs (hereafter DNN-relu). And the third model was a DNN with sigmoid activation functions trained with the CE criterion during 45 epochs (hereafter DNN-sigm).

Two models belong to the second set of acoustic models. Both take as input FBANK features with a window size of 11 and share the same topology. It consist of one convolution layer followed by a max pooling operation, 6 feed-forward hidden layers of 2048 units each, and an output layer of 10492. The convolutional layer is composed of 128 filters with a filter size of 9 and shift of 1. Meanwhile, the max-pooling layer was configured with a pooling width and shift of 2. The difference between both models is the type of activation functions used for the feed-forward layers: sigmoid (CNN-sigm) and rectified linear (CNN-relu).

4.1.3. DNN Speaker Adaptation

The output from the second recognition step was used to carry out speaker adaptation of DNNs (as indicated at the lower box of Fig. 1). The technique used consisted of a conservative training approach, using a very small learning rate and early stopping [9].

Moreover, we made use of confidence measures at word level to exploit inexpensive yet reliable unsupervised speech data. Specifically, confidence measures are estimated from the output of the second recognition pass in order to improve the DNN adaptation step. Although there are many different ways to estimate confidence measures, here we will resort to the conventional approach by which these measures are computed as word posterior probabilities [10].

In order to take advantage of confidence measures, we decided to use them to weight the samples during the adaptation. In this approach, all samples are taken into account, but the contribution of each sample is weighted by its corresponding confidence measure. The rationale behind this method is that only samples with high confidence measures are relevant for the adaptation process, whereas those with low confidence can be neglected. In some way, this method can be seen as a refinement of taking away those samples behind an specified threshold, avoiding the need of estimating that threshold.

Formally, adaptation with weighted samples is based on a modified cross entropy training criterion:

$$\sum_{n=1}^{N} c_n \log p(s_n \mid \mathbf{x}_n)\,, \tag{2}$$

where $\mathbf{x}_1^N$ is the set of frames, s_n is the senone (label) according to the output from the second pass, and $c_n \in [0, 1]$ is its confidence measure. This modified criterion leads to a different way to estimate errors in the Back-Propagation algorithm. In particular, the error for the nth frame δ^n is

Table 1: Stats of the different LM training corpora. The poliMedia [11], VideoLectures.NET and VL.NET subtitles [12] corpora were generated during transLectures project.

Corpus	Sentences	Words	Perplexity
Europarl	2.2M	53M	454.3
Europarl TV	128K	1.2M	454.5
Giga 10^9	22M	557M	296.9
Google Ngrams	-	303B	1871.1
NewsCrawl	53M	1.1B	151.7
poliMedia	4K	95K	1393.1
VideoLectures.NET	5K	127K	871.4
VL.NET subtitles	85K	1.7M	371.5
Wikipedia	82M	1.5B	200.1
TED train	520K	3.7M	218.2

estimated as follows

$$\delta^n = (\mathbf{y}_n - \mathbf{s}_n) \cdot c_n \,, \tag{3}$$

where $\mathbf{y}^n$ is the output of the last layer, and $\mathbf{s}^n$ are the target labels.

4.2. Language Modeling

We used several different text corpora to train the language models. They were preprocessed to normalise capitalisation, remove punctuation marks and transliterate numbers. We can distinguish two different types of corpora, out of domain corpora (OOD), most of them, and in domain corpora (ID), in this case only TED train set. Table 1 summarises the main figures of all the corpora used.

The vocabulary for the language models have been obtained by selecting the 200K most frequent words of a 1-gram LM interpolation of the OOD corpora. The words form the ID corpus are added to this selection, obtaining a final vocabulary of 209 660 words.

With this vocabulary, we trained standard Kneser-Ney smoothed n-gram models for each one of the corpora using the SRILM toolkit [4]. The order of each model is adjusted to 3 or 4 depending on the size of the corpus. The last column of Table 1 shows the perplexity obtained with all these models on the English development set.

All the resulting models are linearly interpolated to obtain a final powerful model adapted to the characteristics of the task, optimising the interpolation weights on the development set [13]. To reduce the size of the final model, it is pruned by removing those n-grams ($n > 1$) whose removal causes (training set) perplexity of the model to increase by less than 2×10^{-10}. This model obtained a perplexity of 126.1.

4.3. Experimental Results

In this section all the recognition experiments performed for the English transcription system are described. Recognition experiments were carried out on the IWSLT 2015 English ASR development and evaluation sets, the statistics of which are shown in Table 2.

Table 2: Statistics of the English ASR development and evaluation sets.

Set	# Talks	Time
tst2013	28	4h:39m
tst2014	15	2h:22m
tst2015	12	2h:25m

Following the IWSLT evaluation requirements, tst2013 was used as development set, tst2014 as progressive evaluation set and tst2015 as evaluation.

The decoding was performed for all the subsystems following the scheme from Fig. 1. The first step was common and its output was used to perform fMLLR speaker adaptation. After that, each subsystem performed the second recognition step, the output of which was used to perform DNN speaker adaptation using confidence measures. Results from these two steps are shown in Table 3.

Table 3: Effect of DNN Speaker Adaptation on each subsystem in terms of WER. Results are shown on tst2013 data-set.

Subsystem	Non-Adapt	Adapt	R. Improvement
DNN-mmi	16.9	16.7	1.2%
DNN-sigm	17.1	16.7	2.3%
DNN-relu	18.5	17.8	3.8%
CNN-sigm	19.4	18.8	3.1%
CNN-relu	18.7	18.0	3.7%

It is worth mention that none of the above results has been subjected to a process of spelling normalisation by means of a global mapping file. As we can observe, the DNN-mmi adaptation has not performed as the rest of system's adaptations. To our knowledge this is because there is not so much room for improvement as occurs in the other systems, and also to the change in the training criterion (from MMI to CE during adaptation).

Finally, a recogniser output voting error reduction (ROVER) algorithm was applied to combine the subsystem's output and further improve the recognition results. The combination weights were estimated based on the development set, giving 2:2:1:1:1 for DNN-mmi, DNN-sigm, DNN-relu, CNN-sigm and CNN-relu. The final scoring results are shown in Table 4. At the time of writing this paper results on the progress test set tst2014 were not provided.

5. German Transcription System

In this section the German ASR system is described. The first section details the data and training procedure, while the second section shows the results obtained by the system.

Table 4: The final results of the English system in terms of WER. (* means official result)

Set	ROVER
tst2013	16.2
tst2015	13.3*

5.1. Training

For the acoustic modelling, we decided not to use the Euronews ASR provided corpus due to processing power constraints and its acoustic conditions being far from target conditions. Instead, we downloaded and processed the German Speechdata Corpus (GSC) [14], an open source corpus recorded and released by the LT and the Teleccoperation group from the Technical University of Darmstadt. This corpus contains 180 different speakers and 36 hours of speech, recorded under controlled conditions with many microphones in parallel. The whole corpus was used as train data. The grapheme-to-phoneme conversion was performed with the help of MaryTTS software [15].

The training procedure for German was the same as the DNN-MFCC used in the English system (Sec. 4.1.2). 48-dimensional MFCC acoustic vectors were extracted and normalised by speaker. A single acoustic model was estimated for German, which consists of a feed-forward DNN with a window size of 11 and 4 hidden sigmoid layers with 2048 neurons each. The output layer features 12237 senones. The network initialisation was performed with the DPT approach, and then the network was trained using the Cross-Entropy error criterion for 10 epochs.

The training and recognition follow the same three-step approach of the English system. An speaker-independent model is used in the first step. The output transcription is then used to perform unsupervised fMLLR adaptation. This second transcription is employed to perform DNN Speaker adaptation (Sec. 4.1.3). In the case of German, no confidence measures have been used for this third step.

The language model for our German system is made up by a standard linear interpolation of 4-gram language models. These models were estimated from different open corpus downloaded from the Internet. The corpora were normalised by lower-casing, removing punctuation marks and transliterating numbers. The corpus statistics after this process can be found in Table 5.

Table 5: Statistics of the German LM corpus.

Corpus	Sentences	Words	Perplexity
Europarl	2M	46M	515.5
News-crawl	135M	2B	352.0
Wikipedia	31M	326M	423.4

When training, the vocabulary was restricted to 200k words, selected with the same procedure described in Section 4.2. The interpolation weights were set to optimise the perplexity of the dev set. In order to improve recognition time, the interpolated model was pruned with a prune factor of 2×10^{-9}. The perplexity of the language model is 290.4.

5.2. Experimental Results

We tested our system on the tst2013 corpus, which was set as the official development corpus of the 2015 challenge. This corpus contains 9 videos from the TEDx website, with varying acoustic conditions. The results are summarised in Table 6. At the time of writing this work results on tst2014 set were not provided.

Table 6: The final results of the German system in terms of WER. (* means official result)

Set	WER
tst2013	43.6
tst2015	43.3*

Unlike the English task, we were not able to obtain state-of-the-art results for the German task. We attribute this result to the lack of relevant in-domain acoustic resources and the simplicity of the approaches employed.

6. Conclusions

In this paper we have described the English and German ASR systems developed for the IWSLT 2015 evaluation campaign. For the first participation of the MLLP group, the presented systems make use of the hybrid approach of HMM-DNN. Particularly, the decoding step of the English system is based on the combination of five different transcription subsystems. Each one built as a three pass recognition system and combining different types of NNs architectures, input features and activation functions. Meanwhile, the German system constitutes our first large scale speech recognition system on this language and it is based on a three pass recognition system with DNN speaker adaptation.

7. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 287755 (transLectures) and ICT Policy Support Programme (ICT PSP/2007-2013) as part of the Competitiveness and Innovation Framework Programme (CIP) under grant agreement no 621030 (EMMA), the Spanish MINECO Active2Trans (TIN2012-31723) research project, the Spanish Government with the FPU scholarship FPU13/06241 and the Generalitat Valenciana with the VALi+d scolarship ACIF/2015/082.

8. References

[1] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[2] M. A. del Agua, A. Giménez, N. Serrano, J. Andrés-Ferrer, J. Civera, A. Sanchis, and A. Juan, "The translectures-upv toolkit," in *Proc. of VIII Jornadas en Tecnologa del Habla and IV Iberian SLTech Workshop (IberSpeech 2014)*, Las Palmas de Gran Canaria (Spain), 2014. [Online]. Available: http://www.mllp.upv.es/wp-content/uploads/2015/04/IberSpeech2014-TLK-camready1.pdf

[3] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Proc. of the SLT*, 2012, pp. 366–369.

[4] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. of ICSLP'02*, September 2002, pp. 901–904.

[5] "TED: Ideas worth spreading," https://www.ted.com.

[6] A. Rousseau, P. Deléglise, and Y. Estève, "Ted-lium: an automatic speech recognition dedicated corpus," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012.

[7] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. of HLT*, 1994, pp. 307–312.

[8] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 24–29.

[9] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. of the ICASSP*, 2013, pp. 7893–7897.

[10] F. Wessel, R. Schlüter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 288–298, 2001.

[11] "poliMedia," https://polimedia.upv.es/.

[12] "Videolectures.NET: Exchange ideas and share knowledge," http://www.videolectures.net/.

[13] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *In Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands: North-Holland, May 1980, pp. 381–397.

[14] S. Radeck-Arneth, B. Milde, A. Lange, E. Gouvêa, S. Radomski, M. Mühlhäuser, and C. Biemann, "Open source german distant speech recognition: Corpus and acoustic model," in *Text, Speech, and Dialogue*. Springer, 2015, pp. 480–488.

[15] 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.

The Heidelberg University English-German translation system for IWSLT 2015

Laura Jehl, Patrick Simianer, Julian Hitschler, Stefan Riezler

Department of Computational Linguistics
Heidelberg University, Germany
{jehl, simianer, hitschler, riezler}@cl.uni-heidelberg.de

Abstract

We describe Heidelberg University's system for English-to-German translation of transcribed TED talks. Our system follows the hierarchical phrase-based paradigm [1]. We only used data allowed within the constrained track. Consistent gains were found using our in-house implementation of automatic source-side reordering, as well as large-scale tuning with a large, lexicalized feature set. We also confirm the success of large class-based language-models.

1. Introduction

We describe the Heidelberg University (`hdu`) submission to the IWSLT 2015 evaluation. We submitted a system for translating transcribed English TED talks into German, using only data permitted within the constrained track. We focus on improving a hierarchical phrase-based system by adding large language models and thousands of sparse, lexicalized features tuned on a large in-domain data set. We further incorporated syntactic knowledge through source-side reordering and k-best rescoring with language models based on syntactic annotations.

The paper is organized as follows: Our baseline setup is described in Section 2. Section 3 then explains our training pipeline and evaluates the contributions of each step. In Section 4, we show that scaling up the feature set and training a parallelized pairwise ranking optimizer on a larger development set further improves our system. We also conduct ablation experiments for different feature templates. Section 5 describes the integration of various external knowledge sources via k-best rescoring.

2. SMT system

All our systems use the `cdec`[1] tools for phrase extraction and decoding [2]. Our language models are estimated using KenLM [3]. For parameter tuning we use our in-house pairwise ranking optimizer `dtrain`, which is available in the `cdec` repository [4]. This section describes data preparation and the baseline system.

2.1. Data

We used all provided bilingual training data. Prior to training, we filtered out empty lines and any pairs containing sentences longer than 150 words. For the common crawl data, we applied an additional filtering step by running `langid.py` [5] on both sides to filter out sentences in the wrong source or target language. Datasets were tokenized with `cdec`'s `tokenize-anything.sh` and truecased using the truecaser available in the Moses toolkit.[2] All systems described in Sections 2 and 3 were tuned on the IWSLT `dev2010` development set with `tst2012` and `tst2013` used as progress test sets. We then added `tst2011-13` to our tuning data (Section 4), leaving `tst2010` as a held-out set for tuning our k-best reranker (Section 5). `tst2014` was treated as a blind test set.

2.2. Baseline

Our baseline model includes 21 features, namely bidirectional lexical phrase pair and word pair probabilities, seven pass-through features, three arity penalty features, a 4-gram language model built from the target side of the training data and count features for word penalty, glue rules, and language model OOVs.

3. Training Pipeline

We now describe our training pipeline and feature set and evaluate their performance of each step. The results are listed in Table 1. All tables report cased, detokenized BLEU scores obtained via the evaluation server provided by the task organizers.

3.1. Source-side reordering

To account for differences in word order, we re-arranged all source-sentences to match the syntax of the target language by applying a variation of the approach described in [6]. This approach works by permuting nodes in a dependency tree. During training, the reorderer generates all possible reordering rules within a window of three nodes governed by the same parent nodes. It then selects the rule which reduces the number of crossing alignments most on a randomly selected

[1] `https://github.com/redpony/cdec`

[2] `http://www.statmt.org/moses/`

validation set. This rule is applied to the training data and the procedure is repeated. Through this repeated permutation, the algorithm is able to generate long-range reorderings. A reordering rule stores part-of-speech and dependency label information of nodes, and a permutation order. If a matching configuration is found at test time, the permutation is applied. In order to reduce training time and to learn rules specific to spoken language, we trained the reorderer on in-domain data only. We used the Stanford parser for English[3], but our implementation can also be applied to the output of other parsers, e.g. in CoNLL format. The code will be made available.[4] We reordered and re-aligned all training data. Source-side reordering produced small but consistent improvements of 0.1 - 0.37 BLEU (experiment 1).

3.2. Domain adaptation

For domain adaptation, we added a 4-gram language model trained on the target side of the WIT3 data only to the log-linear model. In addition to that, we annotated each hierarchical phrase with binary features indicating which corpora it came from, allowing the model to learn a log-linear scaling weight for this phrase. This approach is similar to the work of [7]. Domain adaptation improved the model by 0.3 BLEU points (experiment 2).

3.3. Sparse alignment features

We included lexicalized alignment indicator features which model word alignment, deletion and insertion in source and target, as described in [8]. Even when tuned on a small development set, these features produced consistent gains of 0.16 to 0.29 BLEU points (experiment 3). More sparse features are described in Section 4.

3.4. Large and class-based language models

Previous work has shown the effectiveness of class-based language models (e.g. [9]). We used `brown-cluster`[5] to infer word classes from the language model training data. Since the KenLM implementation of class-based language models uses as an additional feature the probability $p(w|c)$ of a class c generating a word w, we normalized the raw frequencies returned by `brown-cluster`. We first trained a 7-gram class-based language model using 50 classes on the target side of the training data (experiment 4), but observed only a small improvement on `tst2012`, and no improvement on `tst2013`.

However, when increasing the size of the monolingual training data for word- and class-based language models to 26.8 million sentences, we were able to improve by 1.4 - 2 BLEU points (experiments 5a and 5b). We first added 300 thousand sentences from German political speeches to the language modelling data. We then applied cross-entropy based data selection using an in-domain language model to select 50% of the sentences from newscrawl, as described in [10]. To avoid the selection bias for shorter sentences, we only selected sentences with 5 words or longer. After de-duplication, we obtained 26.8 million sentences. We then built a 5-gram word-based language model, and a 7-gram class-based language model using 200 classes. We also increased the order of our in-domain language model from 4 to 5.

3.5. Comparing fast_align and GIZA++

To allow faster development, we first trained models using the re-parametrized IBM Model 2 implementation in `cdec` (`fast_align`[6]). However, our experiments confirmed that training alignments with GIZA++ [11] (we used the parallel implementation in `mgiza++` [12]) gave a significant boost in performance of 1.01 up to 1.6 BLEU (experiment 6), similar to the discrepancies observed in [13]. In particular, we observed that GIZA++-alignments substantially increased the number of extracted phrases: On `dev2010`, GIZA++-alignments produced 3.2 times as many phrases as `fast_align`.

4. Large-scale tuning with sparse features

Due to the successful results with the sparse alignment features we experimented with a wider range of sparse features (all implemented in `cdec`):

- rule identity features: one binary feature per rule
- rule shape features: generalized rules, by mapping to sequences of terminal and non-terminals
- rule bigram features: all bigrams of terminal and non-terminals inside rules, in both source and target sides

In addition to the plain rule shape features, in which terminals are replaced by a single terminal token, we also apply a variant where terminals are replaced by their word class.

While the rule identity features virtually allow to re-train the full grammar in tuning by assigning individual weights to every rule, rule shape and bigram features assess the quality of certain extraction patterns.

In total, the number of potential features is extremely high, several magnitudes larger than the total size of the grammar.

4.1. Online pairwise ranking optimization

Pairwise ranking optimization for SMT [14, 15, 4] allows tuning of a large number of features, in contrast to the traditional minimum error rate training [16]. We employ an online variant of this training scheme [4] with data sharding,

[3] `http://nlp.stanford.edu/software/lex-parser.shtml`

[4] `http://www.cl.uni-heidelberg.de/statnlpgroup/software.mhtml`.

[5] `https://github.com/percyliang/brown-cluster`

[6] `https://github.com/clab/fast_align`

exp	model	tst2012	Δ	tst2013	Δ	tst2014	Δ
0	baseline	19.78	–	21.38	–	18.86	
1	+ source-side reordering	20.15	+0.37	21.48	+0.1	19.03	+0.17
2	+ domain adaptation	20.46	+0.31	21.76	+0.28	19.32	+0.29
3	+ lexical alignment indicators	20.63	+0.17	22.05	+0.29	19.48	+0.16
4	+ class-based 7-gram (small, c=50)	20.91	+0.28	22	-0.05	19.59	+0.09
5a	+ large word-based LM (26.8M sentences)	21.34	+0.43	23.28	+1.28	20.28	+0.69
5b	+ large class-based LM (c=200)	22.05	+0.71	24.07	+0.79	20.64	+0.36
6	+ GIZA++	23.06	+1.01	25.59	+1.52	22.24	+1.6

Table 1: Components of the training pipeline.

feature selection by $\ell_1\ell_2$ regularization and randomization of the training input [17].

Sharding of the data greatly improves efficiency, as the tuning and optimization may run on several parts of the data at once. The models of different shards can then be mixed via simple averaging. Additionally, we use $\ell_1\ell_2$ regularization with a simple cut-off at 100,000 features per iteration. The input is randomized to counter-act potential effects which would depend on the order of the data. The shard size was chosen to reflect the typical tuning set size of about 1,000 segments.

The final model is an average of the weight vectors of all (15) training iterations. Longer training time neither lead to further improvements, nor did the model overfit. As the algorithm is a (margin) perceptron at its core, it has a single meta-parameter η which can be interpreted as a learning rate. Its optimal value 10^{-4} was found by a simple grid search. Note, that starting from $\mathbf{0}$, a fixed learning rate has no effect on the final model. With the margin perceptron however, it serves as a scaling factor which implicitly controls the number of pairs considered for each k-best list.

An ablation test, concatenating `dev2010`, `tst2011` and `tst2012` for tuning and validating on `tst2013` is given in Table 2. The baseline (experiment 8a) uses 27 features, including a single language model and the domain features. Isolating features shows some notable results (experiments 8b, 8c): While rule identifiers slightly degrade below the baseline (8b) and bigram and shape features show only little improvement (8c), the combination of bigram and shape features can be improved using ids by about 0.4 points (8d). A similar behavior can be observed with the lexical alignment indicators. When combining more sparse feature templates (experiment 8d), the final model sizes are very similar, as are the results on the validation set, which implies no or just a small additivity of lexical and rule id features. Improvements however are best at 0.74 points combining all features (experiment 8e). For the baseline system three runs were carried out to test the effect of the randomization – the standard derivation of the final score is quite low at 0.06 points.

When applying large-scale tuning with all features to our best setup from Section 3, we obtained a further improvement of 0.5 - 0.92 BLEU points (Table 3, experiment 9).

Exp.	feature set	tst2013 Δ	model size
8a	baseline	23.14±0.06	27
8b	bigram	23.44 Δ+0.30	150,514
	lexical	23.49 Δ+0.35	69,105
	id	22.99 Δ−0.15	224,685
	shape	23.30 Δ+0.16	202,777
8c	lex., id	23.15 Δ+0.01	227,743
	bigram, shape	23.37 Δ+0.23	204.537
8d	lex., id, shape	23.56 Δ+0.42	272,061
	bigram, id, shape	23.73 Δ+0.59	265,316
	bigram, lex., shape	23.77 Δ+0.63	228,929
	bigram, lex., id	23.81 Δ+0.67	280,830
8e	all	23.88 Δ+0.74	260,697

Table 2: Ablation test for sparse features (the baseline used GIZA++ alignments, but only one target-side 4-gram language model).

5. k-best rescoring with syntactic and neural network language models

We incorporated more knowledge sources via k-best rescoring. We used three in-domain language models built from target side syntactic annotation, namely part-of-speech, morphology and lemma. The annotations were obtained by running the German dependency parser `parzu`[7]. We also trained an in-domain and a target-side feed-forward neural language model using the NPLM toolkit [18]. All experiments used $k = 100$.[8]

Weights for the different language models were learned using a pairwise ranking approach as described in [19], with an SGD classifier from `scikit-learn` [20]. We did not re-tune the SMT model features, but instead used the model score as a single feature to be tuned.

Results for rescoring are given in Table 3. The first two entries (experiment 4 and 7) show results for the best small-scale system described in Section 3 (no large language models, word alignments from `fast_align`). For this system, we observed gains on `tst2012` and `tst2013`, but a small

[7] `https://github.com/rsennrich/parzu`

[8] We experimented with $k = 1000$, but did not see an improvement.

loss on `tst2014`. The two bottom entries (experiment 9 and 10) show the effect of k-best rescoring on our best system, including large language models, GIZA++ alignments and large-scale tuning as described in the previous section. With this seutp, rescoring did improve BLEU. However, we conducted a small-scale human evaluation by having four raters express pairwise preferences for 30 randomly chosen sentences. The translation pairs were permuted and presented in different order to each rater. In total, we observed a preference for the rescored system in 61.67 percent of the cases, with an average pairwise agreement of 0.36 between annotators. This lead us to still submit the rescored system as our primary submission with the system without rescoring as contrastive submission.

6. Conclusions

We built a hierarchical phrase-based translation system for English-German translation using source and target side syntactic information, large-scale class- and word-based language models, and large-scale tuning with sparse features. On the small scale, combining source-side reordering, domain adaptation, sparse lexicalized alignment features, and a class-based language model, yielded 0.62 - 1.13 BLEU over our baseline. We were able to gain 1.14 - 2.07 BLEU points by adding large language models. Using slower, but more reliable, GIZA++ training, another 1.01 - 1.52 BLEU points were gained. Large-scale tuning with sparse features gave a further 0.5 - 0.92 BLEU points. For k-best reranking we observed gains on the held-out sets for the smaller model, but no additional gains in BLEU over the large model. However, human evaluation indicated a preference for the reranked outputs. Our final results are stated in Table 3 (experiment 9 and 10). They exceed the official baseline by 4.73 - 5.88 BLEU.

7. References

[1] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, 2007.

[2] 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.

[3] 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, 2013.

[4] 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*, Jeju, Korea, 2012.

[5] M. Lui and T. Baldwin, "`langid.py`: An off-the-shelf language identification tool," in *Proceedings of the ACL 2012 System Demonstrations*, Jeju Island, Korea, 2012.

[6] D. Genzel, "Automatically learning source-side reordering rules for large scale machine translation," in *Proceedings of the 23rd international conference on Computational Linguistics*, Beijing, China, 2010.

[7] J. Niehues and A. Waibel, "Domain adaptation in statistical machine translation using factored translation models," in *Proceedings of EAMT*, Stroudsburg, PA, USA, 2010.

[8] F. Hieber and S. Riezler, "Bag-of-Words Forced Decoding for Cross-Lingual Information Retrieval," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies*, Denver, Colorado, 2015.

[9] J. Wuebker, S. Peitz, A. Guta, and H. Ney, "The RWTH Aachen Machine Translation Systems for IWSLT 2014," in *Proceedings of the Int. Workshop on Spoken Language Translation*, South Lake Tahoe, CA, USA, 2014.

[10] 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*, Edinburgh, Scotland, 2011.

[11] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, 2003.

[12] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, Columbus, Ohio, USA, 2008.

[13] C. Ding, M. Utiyama, and E. Sumita, "Improving fast_align by reordering," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015.

[14] L. Shen, A. Sarkar, and F. J. Och, "Discriminative reranking for machine translation," in *HLT-NAACL*, 2004, pp. 177–184.

[15] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki, "NTT statistical machine translation for IWSLT 2006," in *2006 International Workshop on Spoken Language Translation, IWSLT 2006, Keihanna Science City, Kyoto, Japan, November 27-28, 2006*, 2006, pp. 95–102.

[16] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Sapporo, Japan, 2003.

exp	model	tst2012	Δ	tst2013	Δ	tst2014	Δ
4	no rescoring	20.91	–	22	–	19.59	–
7	+ rescoring	21.25	+0.34	22.86	+0.86	19.4	-0.19
9	Contrastive (large-scale, no rescoring)	23.98†	–	26.09	–	23.24	–
10	Primary (large-scale + rescoring)	23.93†	-0.05	25.97	-0.12	23.22	-0.03

Table 3: Reranking experiments. † indicates different tuning sets: This experiment was tuned on dev2010, tst2011 and tst2013, leaving out tst2012.

[17] P. Simianer and S. Riezler, "Multi-task learning for improved discriminative training in SMT," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, 2013.

[18] 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.

[19] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., 2011.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, 2011.

The LIUM ASR and SLT Systems for IWSLT 2015

Mercedes García-Martínez, Loïc Barrault, Anthony Rousseau,
Paul Deléglise, Yannick Estève

LIUM, University of Le Mans, France
firstname.lastname@lium.univ-lemans.fr

Abstract

This paper describes the Automatic Speech Recognition and Spoken Language Translation systems developed by the LIUM for the IWSLT 2015 evaluation campaign. We participated in two of the proposed tasks, namely the Automatic Speech Recognition task (ASR) in German and the English to French Spoken Language Translation task (SLT). We present the approaches and specificities found in our systems, as well as our results from the evaluation campaign.

1. Introduction

This paper describes the ASR and SLT systems developed by the LIUM for the IWSLT 2015 evaluation campaign. We participated in the two tasks mentioned above, with German language for the ASR task; and English to French for the SLT task.

The remainder of this paper is structured as follows: in section 2.1, we describe the data used for both tasks and how the selection was performed. In section 2, we present the architecture of our ASR system and the results obtained on the various corpora used during the campaign. Then in section 3, we expose the architecture of our SLT system, along with its results during the campaign. Lastly, the section 4 concludes this system description paper.

2. Automatic Speech Recognition Task in English

In this section, we will describe the Automatic Speech Recognition system developed by the LIUM for the IWSLT 2015 campaign, as well as present the results (both in-house and official) obtained on various corpora.

2.1. Data selection for the ASR task

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.

2.1.1. Data selection for acoustic models training

For our acoustic modeling we used as a primary source the Euronews ASR 2013 dataset [1] provided by the campaign organizers. In order to strengthen this base, we added data from various in-house sources. Then, we also collected a set of TEDx talks in German and carefully removed the off-limit talks. The Table 1 summarizes the characteristics of the data included in our ASR system acoustic models.

Corpus	Duration	Segments	Words
Euronews	69.1h	22 707	506 019
In-house	207.2h	42 316	2 018 262
TEDx	38.0h	42 633	312 142
Total	314.3h	107 656	2 836 423

Table 1: Characteristics of the acoustic data used in the LIUM ASR system acoustic models.

2.1.2. Data selection for language models training

Since language models training data is constrained for the ASR task, we applied our data selection tool XenC [2] on each allowed corpus at our disposal: basically all of publicly available WMT15 data and a collection of TEDx Talks closed-captions. 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. The table 2 summarizes the characteristics of the monolingual text data used to estimate our system language models.

2.2. Architecture of the LIUM ASR system

Our architecture is based on two separate systems, mainly based on the Kaldi open-source speech recognition toolkit [3] which uses finite state transducers (FSTs) for decoding. A first pass is performed by using a 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 3-gram, then a 4-gram back-off LM and

Corpus	Original # of words	Selected # of words	% of Orig.
IWSLT14	2.85M	2.85M	100.00
Common Crawl	48.04M	4.24M	8.82
Europarl	47.40M	3.20M	6.74
News Crawl	1.4G	130.60M	9.26
News-Comm.	5.06M	0.62M	12.25
Total (w/o IWSLT14)	1.5G	138.66M	9.18

Table 2: Characteristics of the monolingual text data used in the LIUM ASR system language models.

5-gram Continuous Space Language Model [4]. Last, an accelerated version of the consensus approach [5], which takes into account temporal information to speed up the processing, is applied on the confusion networks built from the 5-gram rescored word-graphs.

2.2.1. 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 314.3 hours set, with 9 500 tied triphone states and 325 000 gaussians.

On top of these models, we train two separate deep neural networks (DNNs). The first one is based on TRAP features: For each frame, DNN inputs were composed of 368 TRAP coefficients computed on a sliding window of 31 frames. Each frame was constituted by the outputs of 23 Mel-scale filterbanks. Speaker adaptation was trivial: it only consists on mean subtraction applied on all frames associated to a speaker. It has been trained on the full 314.3 hours set. The DNN was built following the approach described in [6] and it was composed of 6 hidden layers with 2048 units, while the output softmax layer had 4627 outputs. The second one is 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 314.3 hours set as the training material. The resulting network is composed of 6 hidden layers with 2 048 units, while the output dimension is 7 827 units and the input dimension is 440, which corresponds to an 11 frames window with 40 LDA parameters each.

To speed up the learning process, each DNN is trained using general-purpose graphics processing units (GPGPU) and the CUDA toolkit for computations.

2.2.2. Language modeling

For language modeling, we rely on the SRILM language modeling toolkit [7] as well as the Continuous Space Language Model toolkit. The vocabulary used in the LIUM ASR systems is composed of 131 435 entries. The language models are trained on the data presented in section 2.1.2 and separate sets are trained for each system.

With the SRILM toolkit, a 2-gram LM is estimated for each corpus source, with no cut-offs and the modified Kneser-Ney discounting method. These 2-gram LM are then linearly interpolated to produce the final 2-gram LM which will be used in the final system, using the German IWSLT 2013 test corpora. To rescore the word-lattices produced by Kaldi, a trigram and a quadrigram language models (namely 3G and 4G) 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. Table 3 and table 4 details the interpolation coefficients for the 2G, 3G and 4G language models as well as the final perplexity for each final model used in the two systems, respectively for the TRAP-based and the fMLLR-based system.

Corpus	Coefficients		
	2G	3G	4G
manual transcriptions of speech	0.21	0.16	0.16
Common Crawl	0.03	0.05	0.05
News Crawl	0.21	0.18	0.17
Europarl	0.04	0.06	0.07
News-Comm.	0.51	0.55	0.0.55
Perplexity	379	279	264

Table 3: Interpolation coefficients and perplexities for the bigram (2G), trigram (3G) and quadrigram (4G) language models used in the LIUM ASR TRAP-based system.

2.3. Word-lattice merging

Both systems used the same audio segmentation, provided by the LIUMSpkDiarization[8] speaker diarization toolkit. 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.

Both systems provide classical word-lattices with usual information: words, temporal information, acoustic and linguistic scores. Before merging lattices, for each edge, these scores are replaced by its *a posteriori* probability. Posteriors are computed for each lattice independently, then weighted

Corpus	Coefficients		
	2G	3G	4G
IWSLT14	0.016	0.014	0.012
Common Crawl	0.028	0.023	0.020
Europarl	0.075	0.090	0.097
News Crawl	0.872	0.866	0.865
News-Comm.	0.008	0.008	0.006
Perplexity	514	349	326

Table 4: Interpolation coefficients and perplexities for the bigram (2G), trigram (3G) and quadrigram (4G) language models used in the LIUM ASR fMLLR-based system.

by $\frac{1}{n}$, where n is the number of word-lattices to be merged (here, $n = 2$). In our experiments, we did not find significant improvements by using more tuned weights.

For each speech segment, the use of weighted posteriors allows to merge starting (respectively ending) nodes from both lattices together into a single lattice in order to process directly with an optimized version of the consensus network confusion algorithm. This optimization reduces very significantly the computation time by managing temporal information during the clustering steps.

2.4. Results

The LIUM ASR system officialy achieved a Word Error Rate score of 17.8 on the 2015 test set, however, at this time of writing, ranks for each participant and full results have not been disclosed, thus we are not able to provide comparisons.

3. Spoken Language Translation Task

In this section, the architecture of our Statistical Machine Translation (SMT) system used in the SLT task is described.

3.1. Architecture of the LIUM SLT system

The SMT system is based on the Moses toolkit [9]. 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, a 5-gram Operation Sequence Model (OSM) [10] have been trained and included in the system.

3.2. Data processing and selection for the SLT task

All available corpora have been used to train the different components of the SMT system. The source side of the bitexts 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, convert capital letters in lowercase and rewrite numbers in letters.

Once the processing performed, monolingual and bilingual data selection has been applied using XenC [2]. For this purpose, the TED corpus has been used as in-domain corpus (to compute in-domain cross-entropy). The development corpus (named *liumdev15*) was used to determine the quantity of data by perplexity minimization. It is composed of the following corpora : dev2010, tst2010 tst2013.

3.2.1. Translation model

The translation models have been trained with the standard procedure. First, the bitexts are word aligned in both directions with GIZA++ [11]. Then the phrase pairs are extracted and the lexical and phrase probabilities are computed. The weights have been optimized with MERT.

3.2.2. Language modeling

The language model is an interpolated 4-gram back-off LM trained with SRILM [7] 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 on *liumdev15* was 67.02.

Besides, two large context CSLM [12] have been trained, each with a different architecture. Those models are used to rescore the n-best list of SMT hypotheses. Table 5 shows

Name	Order	Proj. size	#hidd. x size	PPL
CSLM11	11	512	3 x 1024	41.98
CSLM19	19	320	3 x 1024	41.38

Table 5: Architecture of the CSLM trained for rescoring the n-best list of SMT hypotheses. The third and fourth columns show the projection layer size and the number and size of the hidden layers, respectively. The last columns contains the perplexities obtained with each model on *liumdev15*.

the details of the architectures of the CSLMs as well as the perplexities obtained on the development corpus *liumdev15*.

3.2.3. Neural network machine translation system

In addition to the phrase-based SMT system, we trained a neural network machine translation (NNMT) system based on [13] during 4 days. It is implemented in the Groundhog framework. It consists in a bidirectionnal encoder-decoder deep neural network equiped with an attention mechanism, as described in Figure 1.

We performed the translation with different values for the beam size. We can observe in Table 6 that the more the beam size is increased, the lower the results in BLEU.

An explanation to this is that the BLEU score differs from the internal score calculated by the model (at the output of the softmax layer). Consequently, a partial hypothesis with

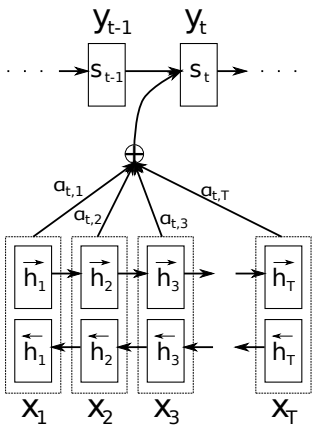


Figure 1: Architecture of the encoder-decoder deep neural network machine translation system equiped with an attention-based mechanism. Taken from [13].

Corpus	*Beam size*		
	10	100	1000
liumtst15	36.79	36.1	35.24
liumdev15	31.62	30.95	30.12

Table 6: Results obtained with the deep NNMT system with different values of beam size.

a low score which is pruned with a small beam size, is kept and extended when the beam size is greater. Moreover, the NN output probability distributions are known to be sharp, giving a high probability to a small number of outputs and a low probability to the rest. This can lead to worse hypotheses having higher results in final. This is an undesirable behavior, which a deeper analysis of the correlation between BLEU score and NN outputs probabilities could explain.

We used the trained NNMT model to rescore the 1000-best list produced by the previously trained SMT model.

3.2.4. Submitted systems

A total of six systems were submitted for evaluation. Several rescoring process have been performed. For the sake of comparison, our best single SMT system has been submitted as *contrastive2* as well as our best NNMT system based on Groundhog (*contrastive5*). This SMT system has been rescored with the two CSLM presented in previous section. *contrastive3* and *contrastive4* correspond to the rescoring with CSLM11 and CSLM19, respectively. Those two systems have also been rescored with the NNMT model obtained with Groundhog. The *primary* system corresponds to the *contrastive3* rescored with Groundhog deep neural network and the *contrastive1* corresponds to the *contrastive4* rescored with the same deep neural translation model.

The results and discussion are presented in the next section.

3.3. Results and discussion

The results obtained on the development and test data are presented in Table 7.

The main observation that we can make is that all the results are coherent. Improvement obtained by rescoring with the CSLM and the NN model on the development corpus are reflected on the internal test (*liumtst15*) and the official evaluation test corpus (*test2015*). The gains observed by rescoring the 1000-best list of hypotheses with a high order CSLM are along previous results in the literature (around +1 BLEU point on development and test data). One can notice that the two different CSLM provide very similar results (in terms of perplexity during training and in terms of BLEU after rescoring).

During system development, we were surprised by the gains observed when rescoring with the NNMT system compared to the lower results obtained (on *liumdev15* and *liumtst15*). An interesting result is that the rescoring with the NNMT model provides similar results on the official test set.

A key point when applying a rescoring process is the optimization of the feature functions weights. The weights for the CSLM and the NNMT model have been optimized with CONDOR [14], a numerical optimizer, with -BLEU as the objective function to minimize. The initial weights are set to those obtained with MERT during the SMT system tuning phase. The initial weights for the CSLM and NNMT features are set to the backoff LM weight (*e.g.* 0.0357). This is motivated by the fact that the LM and CSLM features have a similar distribution. After optimization, the LM had its weights decreased to 0.0314, the CSLM weight increased to 0.0391 while the NNTM feature function saw its weight highly increased (0.0486).

4. Conclusion

We presented the LIUM's ASR and SMT systems which participated in the ASR and SLT tracks of the IWSLT'15 evaluation campaign.

For ASR, we participated to the German transcription task, which is a new challenge to us since we built our first German systems for the campaign. We achieved an official WER of 17.8 of the 2015 test set which seems consistent with our experiments on previous development and test sets.

By rescoring with a continuous space language model, we obtained a gain of about 0.6% BLEU on the SLT test data. On top of that, an additional gain of almost %1 BLEU point is obtained by rescoring with a neural network translation model. The latter result is more surprising since the translation score of the NNMT system is significantly lower than the SMT systems.

5. 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

Name	CSLM rescoring	NNMT rescoring	*liumdev15*	*liumtst15*	*test2015*			
					Case		No-Case	
			%BLEU	%BLEU	%BLEU	%TER	%BLEU	%TER
Primary	CSLM11	yes	33.81	39.61	18.51	79.06	20.02	76.41
Contrast1	CSLM19	yes	33.82	39.65	18.53	78.96	20.10	76.29
Contrast2	-	no	31.81	37.35	16.95	80.61	18.36	78.01
Contrast3	CSLM11	no	32.81	38.36	17.54	80.04	19.02	77.31
Contrast4	CSLM19	no	32.70	38.28	17.56	80.07	19.03	77.45
Contrast5	-	-	31.62	36.79	14.88	84.69	16.98	80.38

Table 7: Results obtained with the submitted systems on internal dev and test corpora and the official evaluation test corpus.

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.

6. References

[1] 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.

[2] 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.

[3] 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.

[4] H. Schwenk, "CSLM - a modular open-source continuous space language modeling toolkit," in *Interspeech*, august 2013, pp. 1198–1202.

[5] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

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

[7] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of Interspeech*, September 2002, pp. 901–904.

[8] S. Meignier and T. Merlin, "LIUM SpkDiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, Dallas (Texas, USA), mars 2010.

[9] 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*, 2007, pp. 177–180.

[10] 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.

[11] 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. [Online]. Available: http://portal.acm.org/citation.cfm?id=1622110.1622119

[12] H. Schwenk, "Continuous Space Language Models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR'15*, 2015.

[14] F. V. Berghen and H. Bersini, "CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm," *Journal of Computational and Applied Mathematics*, vol. 181, pp. 157–175, September 2005.

The UMD Machine Translation Systems at IWSLT 2015

Amittai Axelrod [2], Ahmed Elgohary [1], Marianna Martindale [3],
Khánh Nguyễn [1], Xing Niu [1], Yogarshi Vyas [1], Marine Carpuat [1,2]

Dept. of Computer Science [1], UMIACS [2] and iSchool [3]
University of Maryland, College Park
marine@cs.umd.edu

Abstract

We describe the University of Maryland machine translation systems submitted to the IWSLT 2015 French-English and Vietnamese-English tasks. We built standard hierarchical phrase-based models, extended in two ways: (1) we applied novel data selection techniques to select relevant information from the large French-English training corpora, and (2) we experimented with neural language models. Our French-English system compares favorably against the organizers' baseline, while the Vietnamese-English one does not, indicating the difficulty of the translation scenario.

1. Introduction

Our goal at the University of Maryland (UMD) for the 2015 IWSLT evaluation campaign was to test our redesigned machine translation (MT) pipeline for different language pairs and data conditions. We selected the French-English and Vietnamese-English tasks, consisting of translating the transcripts of TED talks.[1] The French-English task is a standard one, with a large amount of available data. On the other end of the spectrum, the Vietnamese-English language pair is a scarce-resource scenario and has not yet received much attention in the Machine Translation community. We translated into English in both tracks, so as to have a larger amount of monolingual data available for training neural language models. Our systems all use a standard hierarchical phrase-based architecture, outlined in Section 2. We describe how we used data selection techniques (Sections 4 and 5) to make the most of the available data (Section 3). We also discuss the impact of neural language models (Section 6) on translation output. Official results on the evaluation test set show that our French-English systems outperformed the organizers' baseline by +0.65 to +1 BLEU, while our Vietnamese-English system were -3 BLEU below the public baseline. We discuss these results in Section 7.

2. Core Machine Translation Architecture

We use the `cdec` [1] machine translation toolkit to build hierarchical phrase-based MT systems [2]. We expected the resulting synchronous context-free grammar (SCFG) phrasal rules to be well suited to modeling both the local reorderings arising from translating French into English, as well as the more complex translation rules needed to map Vietnamese – an analytic head-initial language – into English. Training the MT systems was done by following the baseline `cdec` pipeline.[2] Word alignments were generated using `fast_align` [3], and symmetrized using the *grow-diag-final-and* heuristic. The SCFG rules extracted for each test sentence were scored using a small number of dense features, including rule frequency, maximum lexical alignment within the rule, *etc*. We mostly used 4-gram language models, trained using `kenlm` [4], unless stated otherwise. Model weights were tuned using the MIRA algorithm [5] in order to maximize BLEU [6] on held-out test sets.

3. Data Preparation

The 2015 IWSLT campaign released parallel data from both Wikipedia [7] and TED talks.[3] The remaining corpora were obtained from the 2015 Workshop on Machine Translation (WMT '15) task.[4] We translated into English in both of the evaluation tracks we participated in. The English data was all pre-processed the same way: first tokenized with the Europarl tokenizer[5] and then lowercased with the standard `cdec` tool.

3.1. French–English Data

We processed the French data in the same way as the English data, described above, except the tokenization was done with the Moses tokenizer. Table 1 lists the specific sources contained in the 41M parallel French-English training corpus.

3.2. Vietnamese–English Data

The Vietnamese-English translation task is a scarce-resource scenario, with only 0.5% as much training data as the French-English task. Our training corpus included all of the parallel data made available by the organizers, including the auto-

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

[2] `http://www.cdec-decoder.org/guide/tutorial.html`

[3] https://sites.google.com/site/iwsltevaluation2015/data-provided

[4] http://www.statmt.org/wmt15/translation-task.html

[5] http://www.statmt.org/europarl/v7/tools.tgz

Proceedings of the 12th International Workshop on Spoken Language Translation
Da Nang, Vietnam, December 3-4, 2015

Corpus	Segments	Tokens (Fr)	Tokens (En)
Europarl v7	2.0 M	61.9 M	55.7 M
News Commentary	200 k	6.3 M	5.1 M
Common Crawl	3.2 M	91.2 M	81.1 M
Gigaword Fr-En	22.5 M	810.2 M	667.9 M
UN Corpus	12.9 M	421.7 M	361.9 M
Wikipedia	403 k	9.8 M	11.3 M
TED corpus	207 k	4.5 M	4.2 M
Total	41.5 M	1.406 B	1.187 B

Table 1: French-English Parallel Training Data

matically extracted Wikipedia corpus [7]. This was done to increase vocabulary coverage, despite the domain mismatch of the Wikipedia data with respect to the TED task. The size of each corpus is shown in Table 2.

Corpus	Segments	Tokens (Vi)	Tokens (En)
TED corpus	130.9k	3.2M	2.6M
Wiki	58.1k	662.2k	661k
Total	189k	3.86M	3.29M

Table 2: Vietnamese-English Parallel Training Data

The processing of the Vietnamese side was minimal: we simply tokenized it as if it were English and removed any uppercasing to normalize borrowed foreign words. We experimented with off-the-shelf chunking tools for Vietnamese, but found that they did not help translation quality. The vnTokenizer [8] tool takes a hybrid approach that combines finite-state automata, regular expressions, and a maximal-matching strategy. However, it proved too slow to process our training data. We also tried the CRFChunker from the JVnSegmenter software [9], which frames chunking as a supervised sequence labeling problem. This tool comes with a model trained on a small set of 8,000 hand-labeled Vietnamese sentences. Unfortunately, using the CRFChunker to preprocess Vietnamese degrades translation quality by about -0.6 BLEU, possibly due to a domain mismatch.

Choosing not to chunk the Vietnamese text differs from standard practice in related translation tasks. In Chinese-English translation, for example, Chinese word segmentation is a key step of the preprocessing pipeline (with the exception of substring or character-based MT models, as in [10]). However, prior work suggests that defining Chinese word boundaries independently of the translation process is not optimal [11, 12]. Based on this, it seems reasonable to let word alignment patterns define translation-driven Vietnamese phrases.

3.3. Postprocessing

Our translation system used tokenized and un-cased data internally. As such, our MT output required the post-processing steps of re-casing and then de-tokenizing before submission. Recasing aims to restore the capitalization that was lost when normalizing case during preprocessing. We used the `Moses` recaser tool.[6] This tool frames recasing as a monotone translation task from un-cased English into cased English. The tool runs Moses without reordering, using a word-to-word translation model and a cased language model. We trained the recaser language model on the English side of the parallel training corpora in Tables 1 and 2. We detokenized the re-cased output using the rule-based detokenizer tool[7] from Moses [13]. We extended this script to support additional special characters that caused the decoder to crash.

4. Training Data Selection

We faced two problems when building the French-English system. The training process was computationally expensive because of the large amount of parallel training data (41M segments). Additionally, the vast majority of the parallel segments are drawn from various domains and genres that are very different from TED. Table 1 shows that TED talks represent only 0.5% of the parallel segments. We addressed both issues by using data selection to determine the most TED-like subset of the parallel corpus. This pseudo in-domain subset was then used to augment the TED data. This approach yielded a medium-scale training setting, easily handled by our standard MT pipeline on ordinary-sized computers.

4.1. Data Selection Techniques

We compared two data selection techniques in the French-English track. The first was the popular cross-entropy difference or "Moore-Lewis" method from [14], which we refer to as `xediff` for short. The second one was recently proposed [15] and uses a hybrid word/part-of-speech text representation to distinguish between rare and frequent events.

4.1.1. Cross-Entropy Difference

The Moore-Lewis method relies on cross-entropy difference to produce domain-specific systems that are usually as good as or better than systems using all available training data [16]. To implement Moore-Lewis selection, we first trained an in-domain language model (LM) on the in-domain TED data, and another LM on the full pool of general data. The algorithm uses these language models to assign a *cross-entropy difference score* to each data-pool sentence.

Lower scores for cross-entropy difference indicate more relevant sentences, namely those that are *most like* the target domain and *most unlike* the full pool average. After ranking the data pool sentences by this score, the top-n sentences (or sentence pairs) are used to create the desired subset of most-relevant sentences. In this work, we added these sentences to the in-domain corpus and trained MT systems on the combined corpus. A range of values for n is typically considered, selecting the n that performs best on held-out

[6] `http://www.statmt.org/moses/?n=Moses.SupportTools`

[7] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/detokenizer.perl`

in-domain data. The size of these domain-specific systems scales roughly linearly with the amount of selected data: a system trained on the most domain-relevant 10% of the full out-of-domain dataset will be roughly one-tenth of the size of a system trained using all the available data.

4.1.2. Hybrid Word/POS Representation

The data selection technique from [15] uses a hybrid word/part-of-speech representation for corpora in order to distinguish between rare and frequent events. In some sense, this newer method is a pre-processing step before performing the above-described cross-entropy difference data selection method. This pre-processing step changes the representation of the corpus into something better suited for computing the relevance score for each sentence. After the sentence scoring and corpus re-ranking is done, the original words are put back and the downstream LM or MT system is trained as usual. This method does not have a standard name yet, so in this work we refer to it as `min10` or `new`.

This newer hybrid word/POS data selection aims to improve scaling of the data selection process itself and to improve the vocabulary coverage of the selected data. This is achieved by constructing a hybrid representation of the text that abstracts away words that are infrequent in either of the in-domain and general corpora. The threshold used to determine "infrequent" is a minimum count of 10 in each of the task and pool corpora, but other values could be explored. All words that do not meet this criterion are replaced with their part-of-speech (POS) tags, permitting their n-gram statistics to be robustly aggregated when the task and pool language models are built.

The intuition for abstracting away rare words is that if a domain-relevant sentence includes a rare word in some non-rare context (e.g. "An earthquake in Port-au-Prince"), then another sentence with the same context but a *different* rare word is probably also just as relevant (e.g. "An earthquake in Kodari"). Suppose "Kodari" is an out-of-vocabulary word with respect to the task corpus, and that "Port-au-Prince" appears three times in each corpus. The cross-entropy difference method would reward the first sentence because it knows "Port-au-Prince", but penalize the second sentence because "Kodari" is unknown. The new method would also reward the first sentence, because it has seen "An earthquake in NPP" a few times. The new method would *also* reward the second sentence, for exactly the same reason.

After the corpus has been transformed, the Moore-Lewis data selection algorithm is then used to select parallel segments on the hybrid corpus representation, the data pool is re-sorted by this score, and then the hybrid corpus representation is discarded and the original representations of the selected segments (the regular sentence forms) are then used to train MT systems.

Recent experiments on medium-scale Chinese-English Machine Translation tasks [15] showed that this hybrid method can substantially improve lexical coverage, reduce computational requirements for the data selection model itself, and improve translation quality when compared against the standard approaches of [14] and [16].

4.2. Training Data Selection Results

Each of the two data selection methods tested for the French-English task has three possible instantiations: as a monolingual method on the input side (French), as a monolingual method on the output side (English), or as a bilingual method that combines both the French and English monolingual scores. In each of the six cases, we selected relevant subsets of the data pool and concatenated each of them with the in-domain TED training data when training the downstream MT system. We used `cdec` to train these downstream systems for extrinsic evaluation.

For consistency, we used the KenLM toolkit [4] to build all language models used for the data selection experiments. All of them were 4-gram LMs. To enable fair comparisons, all of the word-based models had vocabularies fixed to: {TED} $\cup$ {Pool minus singletons}. In constructing our hybrid word/POS representations for the new method, we used the Stanford part-of-speech tagger [17] to generate the POS tags for each of the languages.

The amount of data selected for each method was determined empirically by training MT systems on the selected slices and comparing the BLEU scores on the `tst2012` and `tst2013` held-out sets. We tested all three conditions for each of the two methods, though here we present only results from using the monolingual English version of the cross-entropy difference and the new hybrid methods. The monolingual English results are shown in Figure 1. The new method provides significantly better coverage of the words in the in-domain corpus than the Moore-Lewis method, and at least as good MT performance. Though the new method's BLEU scores are slightly better, the difference is not enough to be particularly important.

For this submission, the best performance with the standard cross-entropy difference method was with 3 million selected sentences. With the new hybrid word/POS method, selecting 4 million sentences out of the 41 M in the data pool. More results, graphs, and detailed analysis comparing the two methods can be found in [18]. The results from the monolingual French and bilingual scoring methods followed the same trend as the monolingual English scores, but were overall slightly lower.

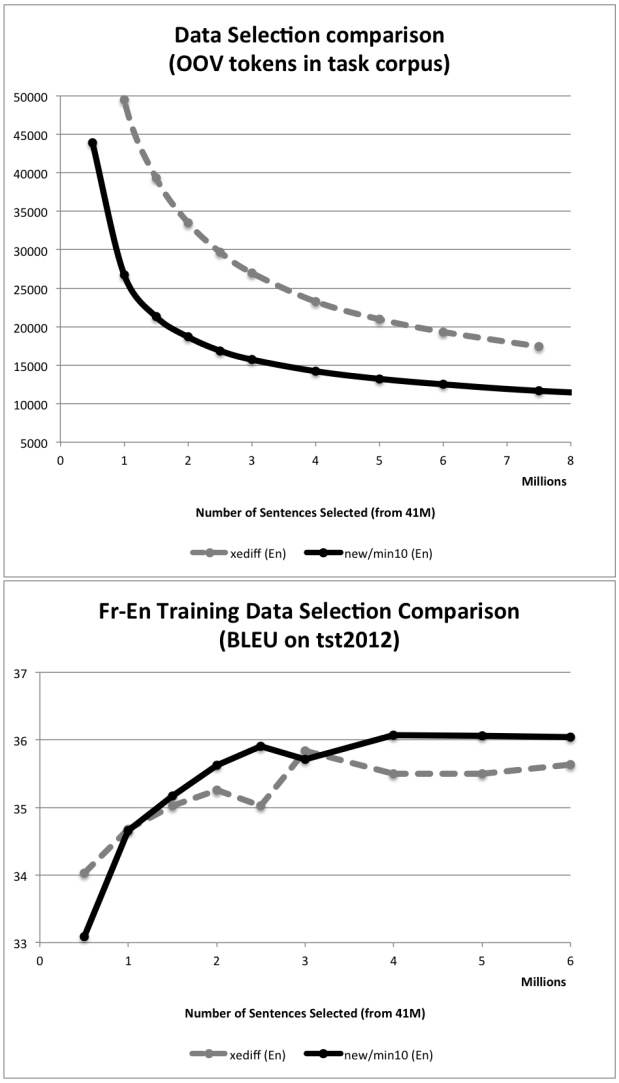


Figure 1: Comparison of the two monolingual English-side data selection methods: Moore-Lewis (grey dashed) and the new hybrid word/POS (solid black): OOV tokens in the TED training set (top), and BLEU scores on `tst2012` (bottom).

Method	BLEU (tst2013)
baseline	37.82
+`xediff` (3 M)	38.29
+`min10` (4 M)	**38.54**

Table 3: Expanding the training set using data selection improves Fr-En translation quality.

After determining the best amount of data to select with each method, we evaluated whether these selected subsets were helpful for translating in-domain test sets. These results are shown in Table 3. The baseline system used the in-domain data in the MT pipeline described in Section 2, and was tuned on the large development set defined below, in Section 5. Table 3 shows that both data selection techniques improve the BLEU score of the translation output. The newer hybrid word/POS method from [15] yielded the largest improvement (+0.7 BLEU), and was therefore used as the training set for our French-English submissions.

5. Tuning Data Selection

Since selecting training data improves translation quality, we hypothesized that similar techniques could also be used to construct better tuning sets. Prior work shows that choosing a good development test set to tune the MT log-linear model parameters is crucial to performance [19, 20]. The IWSLT organizers provided a large number of development test sets for tuning and development purposes (Tables 4 and 5). As a result, we had many options for defining the tuning and tests sets for our experiments.

Corpus	# segments	# fr tokens	# en tokens
dev2010	887	20214	20214
tst2010	1664	33846	31979
tst2011	818	15628	14498
tst2012	1124	23460	21473
tst2013	1026	23293	21706

Table 4: French-English Development Test Sets

Corpus	# segments	# vi tokens	# en tokens
dev2010	769	20750	17410
tst2010	1342	35320	28317
tst2011	1435	32801	26887
tst2012	1553	34292	27983
tst2013	1268	33682	26728

Table 5: Vietnamese-English Development Test Sets

We made the assumption that the most recent test sets would be closest to this year's evaluation data, and therefore used the `tst2013` test set to evaluate translation quality during system development. We proposed two ways to make use of the remaining data at tuning time: First by increasing the number of tuning examples, and secondly by ranking the tuning set and ordering the examples from easiest to hardest.

The development test sets could be used differently: for instance, we could have used several held-out test sets to guide system development.Given our focus on data selection, we decided instead to build a large tuning set by concatenating all development test sets, aside from `tst2013`. As shown in Table 6, this simple strategy yielded a +0.8 BLEU improvement for the Vietnamese-English task, and a +0.75 improvement for the French-English task.

Next, we investigated the impact of ranking the tuning examples. The order in which tuning examples are seen has an impact on learning, because we tune parameters using the *online* MIRA algorithm [21]. Instead of using the natural order of sentences in the original documents, we hypothesized that presenting "easy" examples before "hard" examples might help learning, as in curriculum learning [22].

Task	Tuning set	BLEU
Vi-En	dev2010	23.52
Vi-En	dev2010+tst2010+tst2011+tst2012	**24.30**
Fr-En	dev2010	36.43
Fr-En	dev2010+tst2010+tst2011+tst2012	**37.19**

Table 6: Impact of expanding tuning set on translation quality (train = TED, test set = `tst2013`)

We defined "easier" and "harder" to mean the tuning sentences were more (and less, respectively) similar to the parallel training data. We used the in-domain language model perplexity as a similarity score over sentences. We trained 4-gram models with modified Kneser-Ney smoothing [23] using `kenLM` [4] on the source side of the in-domain TED training data. We then ranked the tuning examples by increasing perplexity. Tables 7 shows that this approach yielded further improvements in translation scores, at least for French-English (+0.6 BLEU), though it had no effect on Vietnamese-English (+0.01 BLEU). This suggests that the order of tuning examples can impact translation quality, but is not guaranteed. However, it is not clear how to best rank examples, and we will investigate alternate ranking criteria (including random order) and re-sampling strategies in future work.

We use the best performing strategy in the final system, and tuned on the concatenation of examples from `dev2010` to `tst2012`, ranked by perplexity.

Task	Tuning set	Order	BLEU
Vi-En	dev2010+tst2010-2012	default	24.30
Vi-En	dev2010+tst2010-2012	ranked	24.31
Fr-En	dev2010+tst2010-2012	default	37.19
Fr-En	dev2010+tst2010-2012	ranked	**37.82**

Table 7: Impact on translation quality of ranking tuning examples by increasing perplexity, for a system trained on the in-domain (TED) data and evaluated on `tst2013`.

6. Neural Language Models

Based on recent promising results[24], neural language models (NLMs) [25, 26] have become standard MT system components. NLMs are typically trained by jointly learning word embeddings and an estimator for the probabilities of words conditioned on their preceding history. We used the Oxford Neural Language Modeling Toolkit (`OxLm`) [27], which implements two useful approximations that can significantly reduce the training and testing time. The first approximation is a class-based factorization to word conditional probabilities where classes are obtained by applying Brown clustering [28] to the vocabulary of the training data. In our experiments, we set the number of clusters to the recommended value of $3\sqrt{|V|}$, where $|V|$ is the vocabulary size. Second, `OxLm` provides an implementation of a noise contrastive estimation (NCE) training algorithm [26] which was shown to dramatically reduce the training time with only a minor reduction to the end-to-end BLEU scores.

We trained two kinds of neural language models on datasets of different scale. The first type (labelled `NlmSmall`) was trained on a small amount of data, with a class-based factorized OxLM using minibatch stochastic gradient descent. The training set consisted of the English side of the in-domain parallel data, described in Section 3. The second set of models (labelled `NlmLarge`) were trained on much larger data sets. These larger corpora were constructed by augmenting the training set from `NlmSmall` wiht subsets of the large pool of permissible monolingual English corpora.[8] We used the `xediff` method described in section 4 to select the 2.5M, 5M, and 7.5M samples from the monolingual pool that were most similar to the training set of `NlmSmall`. We trained three class-based factorized OxLms, one for the concatenation of each selected subset with the `NlmSmall` training corpus. These models are labelled `NlmLarge2.5m`, `NlmLarge5m` and `NlmLarge7.5m` in Table 8. We used the NCE-based algorithm to speed up the training of the three large models.

Neural LM Model	Hyperparameters	Vi-En BLEU
None (baseline)	N/A	24.23
`NlmSmall`	l:100,h:8,f:15,λ:1	25.23
`NlmLarge2.5m`	l:100,h:6,f:20,λ:2	25.43
`NlmLarge5m`	l:100,h:6,f:20,λ:1	25.29
`NlmLarge7.5m`	l:100,h:6,f:20,λ:1	**25.48**

Table 8: The best hyperparameters and the corresponding BLEU scores of the Vietnamese-English pipeline of each of our neural language models.

We fine-tuned the hyperparameters of our language models based on the devset perplexity of each hyperparameter setting. We considered all combinations of the following values of four hyperparameters: (1) dimension of word embeddings $l = \{50, 100, 200, 300\}$, (2) history length (order) that the model conditions on $h = \{4, 5, 6, 7, 8\}$, (3) frequency cutoff (the frequency threshold below which a word is considered unknown) $f = \{5, 10, 15, 20\}$, and (4) training regularization parameter $\lambda = \{0.01, 0.1, 1, 2, 5\}$. We noticed that setting l to 200 or 300 hurt the training and testing times significantly without introducing much benefit to the perplexity scores. Table 8 shows the final hyperparameters learned.

Finally, we evaluated the impact of the neural language models on the output scores of our Vietnamese-English system. All models improved the BLEU score. The largest improvement (+1.2) was obtained with `NlmLarge7.5m`, which we included in our final Vietnamese-English submission. For the French-English system, we used `NlmSmall`.

[8] `https://sites.google.com/site/iwsltevaluation2015/data-provided`

7. Conclusion

We have described the UMD systems submitted to the IWSLT 2015 evaluation campaign. Official results on the evaluation data are provided in Table 9. This table contains scores on the cased, detokenized, output, unlike our internal experimental results in Sections 4, 5, and 6.[9]

System	vi-en	fr-en (2014)	fr-en (2015)
Primary submission	21.57	33.20	32.59
Organizers' baseline	24.61	32.22	31.94

Table 9: Results on evaluation test sets; BLEU scores are computed on cased, untokenized data, using the official IWSLT evaluation server.

The French-English system outperformed the organizers' baseline by approximately +1 BLEU on the 2014 progress test set, and +0.6 on the 2015 test set. This reiterates the benefits of data selection. It is worth noting that these results were obtained using a single n-gram English language model, trained only on the English side of the parallel corpus.

The Vietnamese-English system performed significantly worse than the baseline. This might be due to the lack of preprocessing on the Vietnamese side: as the Vietnamese text was not segmented, the source context captured in SCFG rules was very narrow. In addition, the English n-gram model was trained only on the English side of the parallel data. This can be problematic in a low-resource task such as Vietnamese-English. After the official evaluation period, we augmented our system with 4-gram language models trained on the monolingual English corpus used for neural language modeling. As expected, this approach improved translation quality: we obtained improvements of up to +2 BLEU points on the development test sets.

Overall, our experiments showed that using a standard MT architecture and focusing on parallel data selection for the task at hand is a simple but effective strategy for building MT systems. We will turn our attention to monolingual English data in future work.

8. References

[1] C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, P. Blumson, H. Setiawan, V. Eidelman, and P. Resnik, "cdec: A Decoder, Alignment, and Learning Framework for Finite-State and Context-Free Translation Models," *ACL (Association for Computational Linguistics) Interactive Poster and Demonstration Sessions*, 2010.

[2] D. Chiang, "Hierarchical Phrase-Based Translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[9] Our internal BLEU scores were all computed using an internal scorer on uncased, tokenized, text.

[3] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," *NAACL (North American Association for Computational Linguistics)*, 2013.

[4] K. Heafield, "KenLM : Faster and Smaller Language Model Queries," *WMT (Workshop on Statistical Machine Translation)*, 2011.

[5] D. Chiang, "Hope and Fear for Discriminative Training of Statistical Translation Models," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1159–1187, Apr. 2012.

[6] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," *ACL (Association for Computational Linguistics)*, 2002.

[7] K. Wołk and K. Marasek, "Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs," *Procedia Technology*, vol. 18, pp. 126–132, 2014.

[8] N. T. M. Huyên, A. Roussanaly, H. T. Vinh *et al.*, "A hybrid approach to word segmentation of Vietnamese texts," in *Language and Automata Theory and Applications*. Springer, 2008, pp. 240–249.

[9] C.-T. Nguyen and X.-H. Phan, "JVnSegmenter: A Java-based Vietnamese Word Segmentation Tool," 2007. [Online]. Available: http://jvnsegmenter.sourceforge.net

[10] G. Neubig, T. Watanabe, S. Mori, and T. Kawahara, "Machine Translation without Words through Substring Alignment," in *ACL (Association for Computational Linguistics)*, 2012.

[11] D. Wu, "Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora," *Computational Linguistics*, vol. 23, no. 3, pp. 377–404, 1997.

[12] P.-C. Chang, M. Galley, and C. D. Manning, "Optimizing Chinese Word Segmentation for Machine Translation Performance," in *WMT (Workshop on Statistical Machine Translation)*, 2008.

[13] P. Koehn, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, C. Moran, C. Dyer, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," *ACL (Association for Computational Linguistics) Interactive Poster and Demonstration Sessions*, 2007.

[14] R. C. Moore and W. D. Lewis, "Intelligent Selection of Language Model Training Data," *ACL (Association for Computational Linguistics)*, 2010.

[15] A. Axelrod, P. Resnik, X. He, and M. Ostendorf, "Data Selection With Fewer Words," *WMT (Workshop on Statistical Machine Translation)*, 2015.

[16] A. Axelrod, X. He, and J. Gao, "Domain Adaptation Via Pseudo In-Domain Data Selection," *EMNLP (Empirical Methods in Natural Language Processing)*, 2011.

[17] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network," *NAACL (North American Association for Computational Linguistics)*, 2003.

[18] A. Axelrod, Y. Vyas, M. Martindale, and M. Carpuat, "Class-Based N-gram Language Difference Models for Data Selection," *IWSLT (International Workshop on Spoken Language Translation)*, 2015.

[19] R. Zbib, E. Malchiodi, J. Devlin, D. Stallard, S. Matsoukas, R. Schwartz, J. Makhoul, O. F. Zaidan, and C. Callison-Burch, "Machine translation of Arabic dialects," in *NAACL (North American Association for Computational Linguistics*, 2012.

[20] A. Matthews, W. Ammar, A. Bhatia, W. Feely, G. Hanneman, E. Schlinger, S. Swayamdipta, Y. Tsvetkov, A. Lavie, and C. Dyer, "The CMU machine translation systems at WMT 2014," in *WMT (Workshop on Statistical Machine Translation)*, 2014.

[21] D. Chiang, "Hope and fear for discriminative training of statistical translation models," *Journal of Machine Learning Research*, vol. 13, pp. 1159–1187, 2012.

[22] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum Learning," *ICML (International Conference on Machine Learning)*, pp. 41–48, 2009.

[23] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–393, oct 1999.

[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," *ACL (Association for Computational Linguistics)*, 2014.

[25] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[26] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," *ICML (Internation Conference on Machine Learning)*, 2012.

[27] 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.

[28] 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.

The KIT Translation Systems for IWSLT 2015

Thanh-Le Ha, Jan Niehues, Eunah Cho, Mohammed Mediani 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 2015 machine translation evaluation. We submitted phrase-based translation systems for three directions, namely English→German, German→English, and English→Vietnamese. For the official directions (English→German and German→English), we built systems both for the machine translation (MT) as well as the spoken language translation (SLT) tracks.

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 discriminative models based on different source-side features and classification methods.

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. In addition to punctuation insertion, we also trained that system for sentence boundary insertion since the SLT's data this year come with no sentence boundary.

1. Introduction

The Karlsruhe Institute of Technology participated in the IWSLT 2015 Evaluation Campaign with systems for English→German, German→English and English→Vietnamese. All systems were submitted for the machine translation track, with additional systems for the spoken language translation track in the official directions (English→German, German→English). This year we participated to the new translation direction: English→Vietnamese and we also conducted a short investigation on the impact of word segmentation in our MT system.

On the translation tasks, we integrated new discriminative word lexicon (DWL) models (section 4). We also featured an innovative rescoring method which allows us to take the whole n-best list into account and scale our systems to many features (section 6). Using this, we could seemlessly integrate plentiful numbers of features including the features from the same category, for examples, different DWL models or different neural network language models (section 5).

For SLT tasks, the handling of ASR input was further refined with sentence boundary insertion using a monolingual translation system called *MonoTrans* (section 3). The *MonoTrans* outperformed the provided baseline system for sentence segmentation.

Our baseline system for all translation tasks will be described in section 2. Following sections will present the focused points of this year's KIT systems. After that, the results of the different experiments for the official MT tasks as well as our English→Vietnamese translation will be reported in details in Section 7, before we summarize our findings in Section 8.

2. Baseline system

Our translation systems were conducted using our in-house phrase-based decoder [1]. In English→German and German→English directions, the parallel sections of TED, EPPS, NC and Common Craw are used while TED is the only corpus that we employed to build the English→ Vietnamese system. Addition to the monolingual parts of those corpora, the English News Discussions and Gigaword data are also included in training German→English language models.

The data is preprocessed prior to training and translation. Exceedingly long sentences and aligned sentence pairs having a big difference in length are removed. Special dates, numbers and symbols are normalized. Smartcasing are applied as well. Compound splitting is also conducted to German source texts following the suggestions of [2]. Word segmentation and other typical preprocessing steps for our English→Vietnamese translation system are investigated in details. In addition, our preprocessing also assure that not all sentences from the corpora are used. The noisy ones from Common Crawl were filtered out by a trained SVM classifier as described in [3].

After preprocessing, GIZA++ Toolkit [4] is utilized to perform word alignments over the parallel data. The alignments are then combined to build the phrase table using Moses toolkit [5]. We use the approach described in [6] to adapt out-of-domain phrase tables into the in-domain phrase table from TED for English→German and German→English systems while no adaptation is applied to the English→Vietnamese one.

In both English→German and German→English systems, 4-gram language models with modified Kneser-Ney

smoothing were trained using the SRILM toolkit [7] and scored in the decoding process using KenLM [8]. For English→Vietnamese direction, a longer context of six words is featured in training and scoring.

In addition to conventional word-based language models, we used other language models which are not based on words but contextual information of words. The bilingual language model, based on a four consecutive pairs of source and target words, is used to increase the bilingual context during translation beyond phrase boundaries as described in [9]. On the other hand, the Part-of-Speech (POS) based language model utilizes morphological information by considering a 9-gram sequence of POS tags. Furthermore, we also used the cluster language model based on series of word classes induced by the MKCLS algorithm [10]. This helps alleviate the sparsity problem of surface words by replacing every word in the training corpus with its word class ID.

In our translation systems, we employ two types of reordering models. The first one performs pre-reoderings on the source side by applying the reordering rules learned from POS information [11, 12] and tree constituents [13]. The POS sequences tagged by TreeTagger [14] are used to produce short- and long-range reordering rules. The parsed trees produced by Stanford Parser [15, 16] are used to perform tree-based reorderings which are proved to be helpful for long-dependency modeling. The resulting reordering possibilities for each source sentence are then encoded in a lattice. The second type is the lexicalized reordering model [17] which stores reordering probabilities for each phrase pair scored from the phrase table and the word alignments produced in previous phases.

Other models, described further in following sections, are integrated into our log-linear framework as features. The corresponding weights of those features are tuned using Minimum Error Rate Training (MERT) against the BLEU score as described in [18].

Some additional features, such as source DWL, neural network-based DWL, neural network-based translation and language models, are incorporated into our systems via the ListNet-based rescoring scheme. We will explain further those features as well as our new rescoring approach later in this paper.

3. Preprocessing for speech translation

Since conventional automatic speech recognition (ASR) systems generate either no or only unreliable punctuation marks and sentence segmentation, we design an additional preprocessing step for the test sets of SLT task. In this step, punctuation marks, segmentation, and case information are augmented using a monolingual translation system [19].

Recently, monolingual translation system has shown good performance in inserting punctuation marks for translating speech data [20, 21]. The importance of having proper sentence boundaries, especially, is more emphasized in the IWSLT evaluation campaign 2015. Unlike the SLT condition of previous years' evaluation campaigns, no sentence boundaries are available. Therefore, we need a system which inserts punctuation marks as well as reliable sentence boundaries.

Following previous research described in [22], we built a monolingual translation system which can also augment sentence boundaries. This preprocsesing will be denoted as *MonoTrans*. We built the *MonoTrans* systems for English and German and applied them to two official SLT tracks, English→German and German→English.

For building the systems, we took the preprocessed source side of the parallel training data (either English→German or German→English) and removed the original sentence boundaries. Instead, we inserted sentence boundaries randomly. Therefore, the models can observe sentence boundaries in various positions. If we use the original corpus as it is, the models will learn to insert a sentence boundary at the end of each sentence. This corpus will serve as the target side data of our *MonoTrans* systems.

In order to create the source side data of the *MonoTrans* systems, we remove all punctuation marks from the data and lowercased all words.

Test data is prepared differently using the shifting window of 10 as described in [22]. In this way, each word can be observed in various contexts. Depending on how often a certain punctuation mark was followed by each word, it is inserted based on an empirically chosen threshold.

For both English and German input data, we used the same models in the *MonoTrans* systems. For training data, we used Europarl, TED, NC, and noise-filtered common crawl data, which sums up to 107 million words for English and 85 million words for German. The alignment between non-punctuated, lower-cased text and punctuated, cased text is obtained from GIZA++ [4].

We used a 4-gram language model built on the entire punctuated data using the SRILM Toolkit [7]. In addition to a bilingual language model [9], a 9-gram part-of-speech-based language model is used. The POS is learned from TreeTagger [14]. Also, a $1,000$-class cluster is trained on the punctuated data. The cluster codes are then used to built the additional 9-gram language model. The models were optimized on the official test set of IWSLT evaluation campaign in 2012.

4. Discriminative Word Lexicon

Discriminative Word Lexicon was first introduced by [23]. DWL estimates the probability of a target word appearing in the translation given the source sentence's words. In the original work, a maximum entropy (MaxEnt) model is trained for every target word to determine whether it should be in the translated sentence or not using one feature per source word.

In [24], the authors extended this conventional DWL with n-gram source and target context features. In this evaluation campaign, however, we use the source context features only since the target context features do not bring any im-

provements in our final system. The model using source context features will be referred to as source-context DWL. The source sentence is represented as a bag-of-n-grams, instead of a bag-of-words. This allows us to include local information about source word order in the model.

In addition to this DWL, we integrated a DWL in the reverse direction in rescoring. We will refer to this model as source DWL. This model predicts the target word for a given source word as described in details in [25].

In a first step, we identify the 20 most frequent translations of each word. Then we build a multi-class classifier to predict the correct translation. For the classifier, we used a binary maximum-entropy classifier[1] trained using the one-against-all approach.

As features for the classifier, we used the previous and following three words. Each word is represented by a continuous vector of 100 dimensions as described in [26].

Using the predictions, we calculated two additional features. The first feature is the absolute number of words, where the translation predicted by the classifier and the translation in the hypothesis is the same. The second feature is the sum of the word to word translation probabilities predicted by the classifier that occur in the hypothesis.

While those DWL models can improve the translation by using local source contexts, they employ MaxEnt classifiers which are linear. Hence, they could not really discriminate well the dependencies among features, e.g. a bigram contains two unigrams which somehow reflect a similar or related semantic feature. On the contrary, non-linear classifiers can model those dependencies better since they have the ability to learn some distinct features on higher abstraction levels. [27] introduces non-linearity into DWL by using a deep architecture of neural networks as the alternative classifier. This is referred as neural network-based Discriminative Word Lexicon (NNDWL) in our system. Furthermore, instead of building an indenpendent MaxEnt model for every target word, using NNDWL could improve the translation because it can be seen as a multi-variate classifier consisting of many classifiers which share information among source and target words.

All the DWL models are trained on TED corpus. As showned in previous work, there is no significant improvement using the DWL models trained on bigger corpora.

5. Neural Network Language Model

The traditional n-gram language model (LM) has been applied successfully in many areas of Natural Language Processing due to its robust and simple principles. However, there are some disadvantages of n-gram LM preventing it to better model the cohesion of texts. One of these disadvantages is that the n-grams are presented in a discrete space, hence, it would be hard to estimate well the probability of unseen n-grams which are semantically related to the n-grams appeared in the training set. Continuous space language models, such as restricted boltzmann machine-based LMs[28] or neural network LMs, have been introduced to solve this problem. Basically, in a neural network LM, the discrete representation of words is linearly transformed to a multi-dimentional continuous space. Then one or two following non-linear hidden layers and a softmax output layer are in charge of the probability estimation of the current word based on the transformed representation of the previous words. The transformation and estimation are jointly learned during training. To reduce the time-consuming calculation of the softmax layer, some advanced structures of the output layer and better training methods are proposed[29, 30].

We experimented with different neural network language model toolkits. We used the Torch framework[2], referred to as NNLM, and the nplm toolkit[3][31], referred to as NPLM, to train a feed forward language model. We used in both cases a context of $n = 8$ and trained the model only on the TED corpus. The scores of those language models were added to the n-best list.

6. ListNet-based MT Rescoring

In order to facilitate more complex models, such as the aforementioned DWL models or the neural network language models, we need some way to integrate them to the baseline scores of the phrase-based system. The natural approach is that we rescored the n-best list of candiates in order to select better translations. Compared to other rescoring methods, we would prefer to take the whole list instead of one or two best candidates, so we implemented the rescorer using the ListNet algorithm [32, 33].

This technique defines a probability distribution on the permutations of the list based on the scores of the log-linear model and one based on a reference metric. Therefore, a sentence-based translation quality metric is necessary. In our experiments we used the BLEU+1 score introduced by [34]. Then the rescoring model was trained by minimizing the cross entropy between both distributions on the development data.

Using this loss function, we can compute the gradient with respect to the weight ω_k as follows:

$$\Delta\omega_k = \sum_{j=1}^{n^{(i)}} f_k(x_j^{(i)}) * \quad (1)$$
$$\left(\frac{\exp(f_\omega(x_j^{(i)}))}{\sum_{j'=1}^{n^{(i)}} \exp(f_\omega(x_{j'}^{(i)}))}\right.$$
$$\left. - \frac{\exp(BLEU(x_j^{(i)}))}{\sum_{j'=1}^{n^i} \exp(BLEU(x_{j'}^{(i)}))}\right)$$

When using the i^{th} sentence, we calculate the derivation by

[1] http://hal3.name/megam/

[2] http://torch.ch/

[3] http://nlg.isi.edu/software/nplm/

summing over all $n^{(i)}$ items of the k-best lists. The k^{th} feature value $f_k(x_j^{(i)})$ is multiplied with the difference. This difference depends on $f_\omega(x_j^{(i)})$, the score of the log-linear model for the j hypothesis of the list and the BLEU score $BLEU(x_j^{(i)})$ assigned to this item. Using this derivation, we used stochastic gradient descent to train the model. We used batch updates with ten samples and tuned the learning rate on the development data. The training process ends after 100k batches and the final model is selected according to its performance on the development data.

The range of the scores of the different models may greatly differ and many of these values are negative numbers with high absolute value since they are computed as the logarithm of relatively small probabilities. Therefore, we normalized all scores observed on the development data to the range of $[-1, 1]$ prior to rescoring.

7. Results

In this section, we present a summary of our experiments for all MT and SLT tasks we have carried out for the IWSLT 2015 evaluation. All the reported scores are case-sensitive BLEU scores calculated based on the provided development and test sets.

7.1. German→English

System	MT		SLT
	Dev	Test	Test
Baseline	26.91	28.69	16.57
+ MKCLS	26.97	29.39	-
+ DWL	27.16	29.67	-
KB Mira Rescoring	26.34	29.61	-
+ sDWL + NNDWL	-	**29.91**	16.89

Table 1: Experiments for German→English (MT)

Table 1 presents the results of our experiments for German→English. `tst2012` and `tst2013` are the development and test sets published by the evaluation organizers. Our baseline system already incorporated a number of advanced models. Reorderings were done using both pre-ordering rules as well as a lexicalized reordering model. We adapted the in-domain and out-of-domain phrase tables using the union candidate selection method. In addition to the large language model trained on all available English data, our baseline used an in-domain language model. A bilingual language model trained on all paralled data was also included in the baseline. When we added a 9-gram in-domain cluster language model trained with 100 word classes, our German→English system gained a 0.7 BLEU point improvement. Using a conventional DWL trained on the in-domain data brought further improvement of almost 0.3 BLEU score. The system at this time was used to produce a list of 300 best translation candidates prepared for rescoring. We tried our rescoring using different strategies such as MERT, PRO, KB Mira and ListNet. The corresponding results on a validation set helped us to choose KB Mira as the best strategy to perform rescoring in this direction. Using this strategy, we rescored the n-best list using the old features and two DWL features from source DWLs (sDWL) and neural network-based DWLs (NNDWL). This achieved our best system with 0.3 BLEU points better than the previous system.

For the spoken language translation tasks, since this year's evaluation does not provide the sentence boundaries, we applied the monolingual translation system for sentence boundary and punctuation insertion as well as smart casing described in the section 3. As a baseline for the task, we used our baseline system from the MT task to translate the SLT texts which are already applied *MonoTrans*. Testing on `tst2013` (after removing all sentence boundaries, puntuations and casing), we got the BLEU scores of 16.57. When we applied our best-performing system from the MT task, the SLT system gained an improvements of 0.32 BLEU scores. We submitted this system as our primary system for German→English SLT task. This system achieved 19.64 BLEU score on the official test set this year (`tst2015`). To show the impact of our sentence boundary and punctuation insertion *MonoTrans*, we also submitted another system as the contrastive one. It is the result that we used our best MT system to translate the official SLT test set in which sentence boundaries and punctuations had been inserted by a baseline system provided by the organizer. This contrastive system has a score of 11.84 BLEU points, 7.8 BLEU points less than our primary system on `tst2015`.

7.2. English→German

We conducted several experiments for English→German translation. They are summarized in Table 2. The development set is the `tst2012` and the test set is the `tst2013` data published by the evaluation organizers. The baseline translation system is a phrase-based translation system using two reordering models mentioned above. The phrase table is adapted from the out-of-domain to in-domain TED data. Word-based and non-word language models such as bilingual, POS-based and cluster language models are integrated in the system. Conventional DWLs using source n-grams are also utilized in this phase. The baseline was tuned by MERT and achieved 25.07 and 26.21 BLEU points for development and test sets, respectively.

We performed the rescoring using the ListNet algorithm described in Section 6 on the n-best translation candidates produced by the baseline system. The features that we used are the scores from source and neural network-based DWL models, as well as the neural network-based language models. Adding source DWLs in rescoring scheme helped to improve the system by around 0.7 BLEU points. The NNDWL gained almost 0.2 BLEU points more. Finally, the neural network-based language models, NNLM and NPLM, in-

creased the performance of our system for more than 0.3 BLEU points, reaching 27.50 BLEU points. This system was submitted as our primary system for English→German.

System	Dev	Test
Baseline	25.07	26.21
ListNet Rescoring	24.27	26.36
+ sDWL	-	26.90
+ NNDWL	-	27.18
+ NNLM + NPLM	-	**27.50**

Table 2: Experiments for English→German (MT)

We participated in the spoken language translation tasks for English→German by translating the output of *Mono-Trans* using our best system in the MT task. We got a score of 16.18 BLEU points on the SLT task's official test set `tst2015`.

7.3. English→Vietnamese

This year the IWSLT evaluation organizers have introduced English→Vietnamese translation task for the first time. From the MT perspective, there are two main problems when translating English to Vietnamese: First, the own characteristics of an analytic language like Vietnamese make the translation harder. Second, the lack of Vietnamese-related resources as well as good linguistic processing tools for Vietnamese also affects to the translation quality.

Vietnamese is an analytic language[4]. There are no inflectional morpheme and only several derivational morphemes. In the contrary, it uses a wide variety of function words, temporal or numerical expressions to reflect the grammatical changes. In the linguistic aspect, we might consider Vietnamese is a morphological-poor language, compared to English, German, Finnish or Arabic. In reality, however, the rich set of pronouns in Vietnamese makes the translation to the language harder.

Another linguistic problem which increases the difficulty of Vietnamese-related translation tasks is that the main word boundary marker in Vietnamese is not white space. White spaces are used to separate syllables in Vietnamese, not words. A Vietnamese word consist of one or more syllables. Thus, like Chinese, Vietnamese text processing tools have to deal with ***Word Segmentation*** problem, i.e. how to determine the word boundaries in Vietnamese texts. Word Segmentation is often the first step to be done in a preprocessing phase in those tools since the basic unit is word, not syllable. In this campaign, we also conducted a short investigation to show the importance of using word segmentation methods in an MT system. It would be helpful for further research work on building such translation systems.

Table 3 shows the development stages of the English→Vietnamese system trained on word-segmented texts. We used vnTokenizer[5] [35] for word segmentation and tokenization. The weights of our phrase-based system were also optimized using MERT on word-segmented texts of `tst2012`. And the reported scores were the BLEU scores when we tested the system on word-segmented `tst2013`.

System	Dev	Test
Baseline	19.04	19.97
+ Prereordering	19.87	20.93
+ BiLM + mkcls	20.03	21.07
+ DWL	20.40	21.42

Table 3: Experiments for English→Vietnamese (MT)

The prereordering using POS-based and Tree-based rules helped the most, improving more than 0.8 BLEU points on the development set and nearly 1.0 BLEU points on the test set. This result was not surprising since Vietnamese and English have large differences in term of word order. Integrating non-word language models, e.g bilingual and cluster LMs, brought slightly improvements on both development and test sets, which were 0.16 and 0.14 BLEU points, respectively. In addition, the system gained further enhancement of 0.35 BLEU scores on the test data when we used source-context DWLs. This was the final system we submitted as the primary to the evaluation.

7.3.1. Word-segmented vs. No word-segmented

To compare our methods trained on word-segmented texts and the texts without word segmentation, we built similar systems trained on those two versions and tested them on a non-segmented independent test set. Table 7.3.1 reports the differences. The Dev* and Test* are the BLEU scores measured on the word-segmented development and test sets, respectively. The others are measured on non-segmented datasets.

On the non-segmentation version, we observed that adding more models into the system always helps. And the effects of the models were quite similar to what we observed in case of word-segmented version. For example, the POS- and tree-based reorderings gained the best improvements and integrating DWL were helpful as well as adding non-word language models. The only exception happened when we conducted lexicalized reordering on the word-segmented version, we noticed a slight degrading in the BLEU scores.

It is interesting to observe that our system trained on the unsegmented version of texts performed better than the one trained on the word-segmented texts. One reason we might use to explain this observation is that the vietnamese word segmentation tool, vnTokenizer, is not good enough for TED data. While it simply brings longer contexts, its quality might

[4] `https://en.wikipedia.org/wiki/Vietnamese_language`

[5] `http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnTokenizer`

System	No Word Segmentation		Word Segmentation		
	Dev	Test	Dev*	Test*	Test
Baseline	24.65	25.66	19.04	19.97	24.95
+ Prereordering	25.55	26.58	19.87	20.93	25.95
+ BiLM	25.58	26.76	19.89	20.99	26.36
+ mkcls	25.77	26.85	20.20	21.12	26.43
+ DWL	25.83	27.18	20.40	**21.42**	26.55
+ Lexicalized Reordering	**25.99**	**27.64**	**20.41**	21.24	**26.62**

Table 4: Experiments for English→Vietnamese

affect the word alignments, which in turn affect to other components in our system. In addtion, the advantages of using longer context in case of training on word-segmented texts can be covered somehow by phrase extraction and language modeling. Since phrases in our MT are basically sequences of words, we can see a phrase in the non-segmented system as a shorter phrase compared to corresponding one in the word-segmented system. We would need a more comprehensive investigation on this problem. Due to the fact that we have been investigating the unsegmented system after the submission deadline, we did not submit the system despite its better performance.

8. Conclusions

In this paper, we described several innovative works that we applied to our translation systems we participated in the IWSLT 2015 Evaluation Campaign. Besides the traditional, official MT and SLT tasks for English→German and German→English, we also submitted the newly published translation tasks English→Vietnamese.

For all official translation directions, we built strong baseline systems including our advanced reordering methods, data selection and adaptation techniques, as well as several word-based and non-word language models. Those individual models proved successful in many of the systems.

The notable enhancement this year is the n-best list rescoring which performed better than other MT optimization techniques and scaled better to a large number of features. We used this rescoring to leverage newly-added features such as the DWLs or other neural language models.

The combination of new features with the traditional features in a rescoring scheme boosted our translation systems in both English→German and German→English direction to more than 1.2 BLEU points improvements. When we applied our techniques for English→Vietnamese, we observed the improvements brought by the individual components. We also showed the effects of using non word-segmented texts in training such a translation system.

A monolingual translation system for punctuation insertion played a vital role in adjusting the ASR output for speech translation. This system was also capable to perform decent sentence segmentation which is necessary for the SLT data this year when they do not have any sentence boundary.

9. Acknowledgements

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 645452.

10. References

[1] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[2] 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.

[3] 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.

[4] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, 2003.

[5] 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.

[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] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *Proceedings of the International Conference on Spoken Language Processing*, Denver, CO, USA, 2002.

[8] K. Heafield, "KenLM: faster and smaller language model queries," in *Proceedings of the 6th Workshop on Statistical Machine Translation*, Edinburgh, United Kingdom, 2011.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] A. N. Rafferty and C. D. Manning, "Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines," in *Proceedings of the Workshop on Parsing German*, 2008.

[16] 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.

[17] 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.

[18] 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.

[19] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling punctuation prediction as machine translation."

[20] T.-L. Ha, J. Niehues, T. Herrmann, M. Mediani, E. Cho, Y. Zhang, I. Slawik, and A. Waibel, "The KIT Translation Systems for IWSLT 2013," in *Proceedings of the International Workshop on Spoken Language Translation*, ser. IWSLT 2013, Heidelberg, Germany, 2013.

[21] 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 International Workshop for Spoken Language Translation (IWSLT 2014)*, Lake Tahoe, CA, USA, 2014.

[22] 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.

[23] 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.

[24] 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.

[25] , "Source Discriminative Word Lexicon for Translation Disambiguation," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT15)*, Danang, Vietnam, 2015.

[26] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations." in *HLT-NAACL*, 2013, pp. 746–751.

[27] T.-L. Ha, J. Niehues, and A. Waibel, "Lexical Translation Model Using a Deep Neural Network Architecture," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT14)*, Lake Tahoe, CA, USA.

[28] J. Niehues and A. Waibel, "Continuous space language models using restricted boltzmann machines." in *IWSLT*, 2012, pp. 164–170.

[29] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language model," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5524–5527.

[30] H. Schwenk, A. Rousseau, and M. Attik, "Large, pruned or continuous space language models on a gpu for statistical machine translation," in *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*. Association for Computational Linguistics, 2012, pp. 11–19.

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

[32] J. Niehues, Q. K. Do, A. Allauzen, and A. Waibel, "Listnet-based MT Rescoring," *EMNLP 2015*, p. 248, 2015.

[33] 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, pp. 129–136.

[34] P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar, "An End-to-end Discriminative Approach to Machine Translation," in *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006)*, Sydney, Australia, 2006, pp. 761–768.

[35] H. P. Le, T. M. H. Nguyen, R. Azim, and T. V. Ho, "A Hybrid Approach to Word Segmentation of Vietnamese Texts," *Language and Automata Theory and Applications*, pp. 240–249, 2008.

The 2015 KIT IWSLT Speech-to-Text Systems for English and German

Markus Müller, Thai-Son Nguyen, Matthias Sperber, Kevin Kilgour, Sebastian Stüker and Alex Waibel

Institute for Anthropomatics
Karlsruhe Institute of Technology
Karlsruhe, Germany

{m.mueller|thai.nguyen|matthias.sperber}@kit.edu
{kevin.kilgour|sebastian.stueker|waibel}@kit.edu

Abstract

This paper describes our German and English *Speech-to-Text* (STT) systems for the 2015 IWSLT evaluation campaign. This campaign focuses on the transcription of unsegmented TED talks. Our setup includes systems from both Janus and Kaldi. We combined the outputs using both ROVER [1] and confusion network combination (CNC) [2] to archieve 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 different sets of permissible 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. For English, our single best system based on Kaldi has a WER of 13.8% on the development set while in combination with Janus we lowered the WER to 12.8%.

1. Introduction

The 2015 *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) [3].

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 English and German ASR systems with which we participated in the TED ASR track of the 2015 IWSLT evaluation campaign. Our English and German systems are based on our previous years' evaluation systems [4]. In addition to our Janus[5] based systems, we also built a system based on Kaldi[6] for English. For this, we used the recipe provided in the Kaldi repository for the TEDLIUM corpus [7]. The Janus 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 8 contains a short conclusion.

2. Data Resources

2.1. Training Data

The following data sources have been used for acoustic model training of our English systems:

a) 200 hours of Quaero training data from 2010 to 2012.

b) 18 hours of various noise data, such as snippets of applause, music or noises from microphone movement.

c) 158 hours of data downloaded from the TED talks website, without disallowed talks.

d) 203 hours of TED talks from the TEDLIUM v2 release [7], excluding disallowed talks.

The Quaero training data is transcribed manually. 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. The TEDLIUM dataset is provided by Laboratoire d'Informatique de l'Université du Maine (LIUM).

For German we used the following data sources:

Data set	# Talks	# Utts	Dur.	Avg. dur.
tst2013 (manual)	28	2246	3.9h	6.3s
tst2013 (auto)	28	2353	4.0h	6.1s
tst2014 (auto)	15	801	2.2h	9.7s
tst2015 (auto)	12	1013	2.2h	7.7s

Table 1: *Statistics of the English development sets ("tst2013") and the English evaluation sets ("tst2014" and "tst2015"), including the total number of talks (*# Talks*), the total number of utterances (*# Utts*), the overall speech duration (*Dur.*), and average speech duration per utterance (*Avg. dur.*). "tst2014" and "tst2015" have been segmented automatically. Properties of the automatic segmentation of "tst2013" are displayed alongside with those of the manual segmentation.*

a) 180 hours of Quaero training data from 2009 to 2012.

b) 24 hours of broadcast news data

c) 160 audio from the archive of parliament of the state of Baden-Württemberg, Germany

For language modeling and vocabulary selection, we used most of the data admissible for the evaluation, as summarized in Tables 2 and 3.

2.2. Test Data

For this year's evaluation campaign, two evaluation test sets ("tst2014" and "tst2015"), as well as development test sets ("tst2013") were provided for both English and German. Table 1 lists these 3 test sets along with relevant properties for English.

All development test sets were used with the original presegmentation provided by the IWSLT organizers. Additionally, "tst2013" has been segmented automatically in the same way as the evaluation test sets.

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 bottleneck features [8] 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). For the extraction of those, we use a pitch tracker [9] and fundamental frequency variation [10]. In [11] 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.

3.1. Deep Bottleneck Features

The use of bottleneck features greatly improves the performance of our GMM acoustic models, but also our Hybrid systems benefit from it as well. Figure 1 shows a general overview of our deep bottleneck features (BNF) 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 [12]. For network training, we used a framework based on Theano ([13], [14]).

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.

4. Automatic Segmentation

In this 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 an approach to automatic segmentation of audio data that is SVM based. This kind of segmentation is using speech and non-speech models, using the framework introduced in [15]. 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 [16]. A 2-phased post-processing is applied for final segment generation.

We generated the segmentations for both English and German using this SVM based segmentation. The parameters for the SVM segmenter were chosen on a per language basis after preliminary experiments.

5. Acoustic Modeling

5.1. Data Preprocessing

For the English TED data in dataset c) 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 [17]. The TEDLIUM data did not require any special preprocessing, except for removing all disallowed talks.

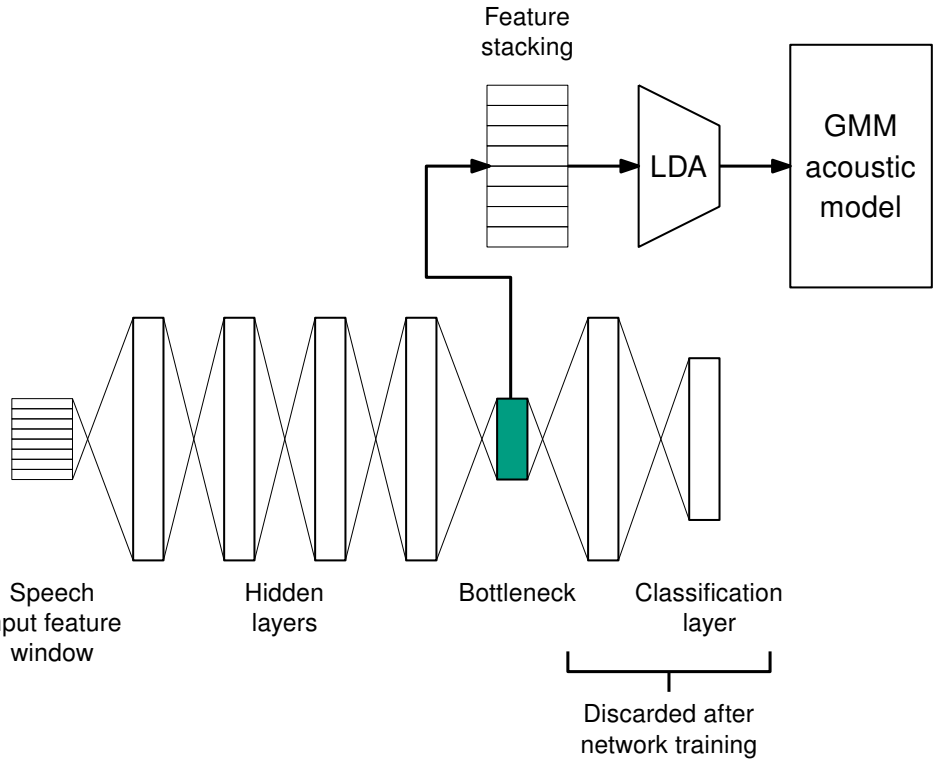


Figure 1: *Overview of our standard DBNF setup.*

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 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) [18], followed by optimal feature space training (OFS) which is a variant of *semi-tied covariance* (STC) [19] training using a single global transformation matrix. The model is then refined by one iteration of Viterbi training. All German models use vocal tract length normalization (VTLN), for English it is used where indicated (V).

In order to improve the performance of our GMM based acoustic models Boosted Maximum Mutual Information Estimation training (BMMIE) [20], a modified form of the Maximum Mutual Information (MMI) [21], is applied at the end. Lattices for discriminative training use a small unigram language model as in [22]. 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.

5.3. Hybrid Acoustic Model

As with the GMM systems we trained our hybrid systems on various 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 15 frames, with 4-5 1600-2000 unit hidden layers and an output layer containing 6016 context dependent phonestates for German and 8156 context dependent phonestates for English. 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.

We trained neural network acoustic models for English on various input features and with different topologies using the same techniques described in the deep bottleneck layer section. Our best setup uses deep bottleneck features stacked over a window of 15 frames, with 5 1600 unit hidden layers and an output layer containing 8156 context dependent phone states. 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 and 14 tonal

features stacked over a 15 frame window.

The German hybrid system is based on a modular topology which involves the stacking bottleneck features from three separate bottleneck extraction networks (MFCC+MVDR+T, lMEL+T & MFCC+MFCC+lMEL+T) over a window of 13 frames leading to a 1638 (=3 * 42 * 13) neuron bottleneck stack, followed by 4 hidden layers containing 2000 neurons each and an output layer containing 6016 context dependent phonestates. The deep bottleneck features were extracted using an MLP with 5 2000 unit hidden layers prior to the 42 unit bottleneck layer. Their inputs were 40 lMel and 14 tone features for the lMEL+T network, 20 MFCC, 20 MVDR and 14 tone features for the MFCC+MVDR+T network and 20 MFCC, 20 MVDR, 40 lMEL and 14 tonal features for the MFCC+MFCC+lMEL+T MLP.

5.4. Kaldi

For system combination we also trained a system using Kaldi [6]. We trained the acoustic model (AM) on the TED-LIUM corpus release 2 [7] using the tedlium receipe (s5). The AM utilizes a neural network taking bottle neck features extracted from combined filterbank and pitch features that are then fMLLR adapted as input. After optimizing its cross-entropy on the training data, the network is refined using sequence training optimizing the sMBR criteria. For the language model we used the cantab-tedlium tri-gram language model [23].

5.5. Pronunciation Dictionary

For English, we used the CMU dictionary[1]. This is the same phoneme set as the one used in last year's systems. It consists of 45 phonemes and allophones. We used 7 noise tags and one silence tag each. For the CMU phoneme set we generated missing pronunciations with the help of FESTIVAL [24].

Our German system uses an initial dictionary based on the Verbmobil Phoneset [25]. Missing pronunciations are generated using both Mary [26] and FESTIVAL [24].

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 and 3). Text cleaning included tokenization, lowercasing, number normalization, and removal of punctuation. Language model training was performed by building separate language models for all (sub-)corpora using the SRILM toolkit [27] with modified Kneser-Ney smoothing. These were then linearly interpolated, with interpolation weights tuned using held-out data from the TED corpus. For German, we split compounds similarly as in [28].

For the vocabulary selection, we followed an approach proposed by Venkataraman et al.[29]. 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, and 300k words for German.

For our English Kaldi system, we used the TEDLIUM language model from Cantab Research[23]. It contains 155,290,779 tokens and is based on the '1 Billion Word Language Model Benchmark'[2].

Text corpus	# Words
TED	3.6m
News + News-commentary + -crawl	4,478m
Euronews	780k
Commoncrawl	185m
GIGA	2323m
Europarl + UN + multi-UN	829m
TEDLIUM dataselection	155m

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.*

7. Decoding Setup

For our English submission we trained 3 different DBNF GMM acoustic models in total by combining different feature front-ends (M2 and lMEL), with and without using VTLN adaptation. We also trained one DNN hybrid system using lMEL front-ends and another one with DBNF features. In addition to these systems, we also included a Kaldi based system this year, using the standard recipe for the TEDLIUM dataset. The first CNC was created using the outputs from 3

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[2] http://www.statmt.org/lm-benchmark

System	tst2013	tst2015	Sub.
lMEL+T+V	17.7	-	
M2+T+V	17.6	-	
lMEL+T	18.1	-	
lMEL+T-DBNF-hyb+V	16.0	-	
lMEL+T-hyb	16.4	-	
CNC 1	14.7	-	
lMEL+T+V+adapt	15.3	-	
M2+T+V+adapt	15.0	-	
lMEL+T+adapt	14.9	-	
CNC 2	14.4	10.9	C 1
Kaldi	15.6	-	
ROVER 1	13.2	-	
Kaldi rescored	13.8	10.4	C 2
ROVER 2	12.8	10.0	Pri

Table 4: *Results for English on 'tst2013' development and 'tst2015' evaluation test sets. Both contrastive systems (**C 1**) and (**C 2**) are shown, as well as the primary submission (**Pri**).*

different DBNF GMM based systems in combination with the output from 2 hybrid systems. Based on this first CNC, the GMM based systems were adapted. Combining the output from the adapted systems and the hybrid systems to another CNC. This second CNC is our first contrastive submission. It contains only output from Janus based systems. The output from our Kaldi setup is incorporated in the first and second ROVER. In the first ROVER, we combined the output from Kaldi, out two hybrid systems and the two best adapted GMM based systems. This result is then included in a second ROVER, where we combined it with the re-scored output from Kaldi and the output from the second CNC. This is our primary condition.

The German setup consists of a DBNF GMM system and a modular Hybrid system. A CNC is performed on the outputs of both systems and used to adapt the DBNF GMM AM. A final CNC is then performed using the adapted GMM output in lieu of the unadapted output.

8. Results

The English systems have been evaluated on the test set "tst2013". The results are listed in Table 4. Based on these results, we decided our decoding strategy for the evaluation. The first CNC results in a WER of 14.7%. Including the output from Kaldi, the WER decreases to 12.8%.

9. Conclusions

In this paper we presented our English and German LVCSR systems, with which we participated in the 2015 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.

10. References

[1] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proceedings the IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, CA, USA, Dec. 1997, pp. 347–354.

[2] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[3] 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.

[4] Kevin Kilgour, Michael Heck, Markus Mller, Matthias Sperber, Sebastian Stker, and Alexander Waibel, "The 2014 KIT IWSLT Speech-to-Text Systems for English, German and Italian," in *International Workshop on Spoken Language Translation (IWSLT)*, Dec. 2014.

[5] M. Woszczyna, N. Aoki-Waibel, F. D. Bu, N. Coccaro, K. Horiguchi, T. Kemp, A. Lavie, A. McNair, T. Polzin, I. Rogina, C. Rose, T. Schultz, B. Suhm, M. Tomita, and A. Waibel, "Janus 93: Towards spontaneous speech translation," in *International Conference on Acoustics, Speech, and Signal Processing 1994*, Adelaide, Australia, 1994.

[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] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks," in *Proc. of LREC*, 2014, pp. 3935–3939.

[8] 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.

[9] K. Schubert, "Grundfrequenzverfolgung und deren Anwendung in der Spracherkennung," Master's thesis, Universität Karlsruhe (TH), Germany, 1999, in German.

[10] K. Laskowski, M. Heldner, and J. Edlund, "The Fundamental Frequency Variation Spectrum," in *Proceedings of the 21st Swedish Phonetics Conference (Fonetik 2008)*, Gothenburg, Sweden, June 2008, pp. 29–32.

[11] 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 *Proceedings of the 10th Workshop on Spoken Language Translation (IWSLT 2013)*, 2013.

[12] 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.

[13] 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, oral Presentation.

[14] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] W. Williams, N. Prasad, D. Mrva, T. Ash, and T. Robinson, "Scaling recurrent neural network language models," *arXiv preprint arXiv:1502.00512*, 2015.

[24] A. Black, P. Taylor, R. Caley, and R. Clark, "The festival speech synthesis system," 1998.

[25] 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.

[26] 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.

[27] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh International Conference on Spoken Language Processing*, 2002.

[28] 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.

[29] 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.

Stanford Neural Machine Translation Systems for Spoken Language Domains

Minh-Thang Luong, Christopher D. Manning

Computer Science Department, Stanford University, Stanford, CA 94305

{lmthang,manning}@stanford.edu

Abstract

Neural Machine Translation (NMT), though recently developed, has shown promising results for various language pairs. Despite that, NMT has only been applied to mostly formal texts such as those in the WMT shared tasks. This work further explores the effectiveness of NMT in spoken language domains by participating in the MT track of the IWSLT 2015. We consider two scenarios: (a) how to adapt existing NMT systems to a new domain and (b) the generalization of NMT to low-resource language pairs. Our results demonstrate that using an existing NMT framework[1], we can achieve competitive results in the aforementioned scenarios when translating from English to German and Vietnamese. Notably, we have advanced state-of-the-art results in the IWSLT English-German MT track by up to 5.2 BLEU points.

1. Introduction

Neural Machine Translation (NMT) is a radically new way of teaching machines to translate using deep neural networks. Though developed just last year [1, 2], NMT has achieved state-of-the-art results in the WMT translation tasks for various language pairs such as English-French [3], English-German [4, 5], and English-Czech [6]. NMT is appealing since it is conceptually simple. NMT is essentially a big recurrent neural network that can be trained end-to-end and translates as follows. It reads through the given source words one by one until the end, and then, starts emitting one target word at a time until a special end-of-sentence symbol is produced. We illustrate this process in Figure 1.

Such simplicity leads to several advantages. NMT requires minimal domain knowledge: it only assumes access to sequences of source and target words as training data and learns to directly map one into another. NMT beam-search decoders that generate words from left to right can be easily implemented, unlike the highly intricate decoders in standard MT [7]. Lastly, the use of recurrent neural networks allow NMT to generalize well to very long word sequences while not having to explicitly store any gigantic phrase tables or language models as in the case of standard MT.

Despite all the success, NMT has been applied to mostly formal texts as in the case of the WMT translation tasks. As such, it would be interesting to examine the effectiveness of NMT in the spoken language domain through the IWSLT MT track. This work explores two scenarios, namely NMT *adaptation* and NMT for *low-resource translation*. In the first scenario, we ask if it is useful to take an existing model trained on one domain and adapt it to another domain. Our findings show that for the English-German translation task, such adaptation is very crucial which gives us an improvement of +3.8 BLEU points over the model without adaptation. This helps us advance *state-of-the-art* results in the English-German MT track by up to 5.2 BLEU points.

For the latter scenario, we show that even with little English-Vietnamese training data, NMT models trained with an off-the-shelf framework can achieve competitive performance compared to the IWSLT baseline. It is also worthwhile to point out a related work [8] which achieved best results for the low-resource language pair Turkish-English in IWSLT. However, their work makes use of a huge monolingual corpus, the English Gigaword.

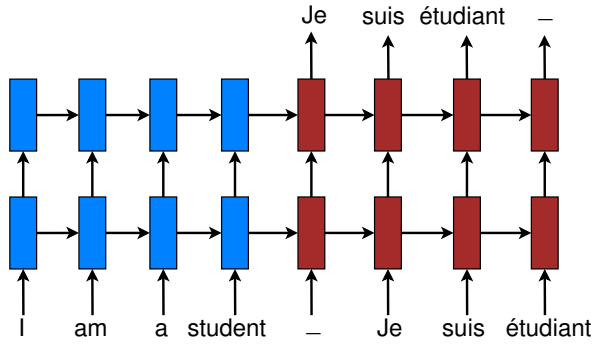


Figure 1: **Neural machine translation** – example of a deep recurrent architecture proposed in [1] for translating a source sentence *"I am a student"* into a target sentence *"Je suis étudiant"*. Here, "_" marks the end of a sentence.

2. Approach

We give background information on NMT and the attention mechanism before discussing our model choices.

2.1. Neural Machine Translation

Neural machine translation aims to directly model the conditional probability $p(y|x)$ of translating a source sentence,

[1] http://nlp.stanford.edu/projects/nmt/

$x_1, \ldots, x_n$, to a target sentence, $y_1, \ldots, y_m$. It accomplishes such goal through the *encoder-decoder* framework [1, 2]. The *encoder* computes a representation $\boldsymbol{s}$ for each source sentence. Based on that source representation, the *decoder* generates a translation, one target word at a time, and hence, decomposes the conditional probability as:

$$\log p(y|x) = \sum_{j=1}^{m} \log p\left(y_j|y_{<j}, x, \boldsymbol{s}\right) \quad (1)$$

A natural choice to model such a decomposition in the decoder is to use a recurrent neural network (RNN) architecture, which most of the recent NMT work have in common. They, however, differ in terms of the RNN architectures used and how the encoder computes the source representation $\boldsymbol{s}$.

Kalchbrenner and Blunsom [9] used an RNN with the vanilla RNN unit for the decoder and a convolutional neural network for encoding the source. On the other hand, Sutskever et al. [1] and Luong et al. [3, 5] built deep RNNs with the Long Short-Term Memory (LSTM) unit [10] for both the encoder and the decoder. Cho et al., [2], Bahdanau et al., [11], and Jean et al. [4, 8] all adopted an LSTM-inspired hidden unit, the gated recurrent unit (GRU), and used bidirectional RNNs for the encoder.

In more details, considering the top recurrent layer in a deep RNN architecture, one can compute the probability of decoding each target word y_j as:

$$p\left(y_j|y_{<j}, x, \boldsymbol{s}\right) = \text{softmax}\left(\boldsymbol{h}_j\right) \quad (2)$$

with $\boldsymbol{h}_j$ being the current target hidden state computed as:

$$\boldsymbol{h}_j = f(\boldsymbol{h}_{j-1}, y_{j-1}, \boldsymbol{s}) \quad (3)$$

Here, f derives the current state given the previous state $\boldsymbol{h}_{j-1}$, the current input (often the previous word y_{t-1}), and optionally, the source representation $\boldsymbol{s}$. f can be a vanilla RNN unit, a GRU, or an LSTM. The early NMT approach [9, 1, 2, 3] uses the last source hidden state $\boldsymbol{s} = \bar{\boldsymbol{h}}_n$ once to initialize the decoder hidden state and sets $\boldsymbol{s} = [\,]$ in Eq. (3).

The training objective is formulated as follows:

$$J = \sum_{(x,y)\in\mathbb{D}} -\log p(y|x) \quad (4)$$

with $\mathbb{D}$ being our parallel training corpus.

2.2. Attention Mechanism

Here, we present a simplified version of the attention mechanism proposed in [11] on top of a deep RNN architecture, which is close to our actual models.

Regarding the aforementioned NMT approach, Bahdanau et al. [11] observed that the translation quality degrades as sentences become longer. This is mostly due to the fact that the model has to encode the entire source information into a single fixed-dimensional vector $\bar{\boldsymbol{h}}_n$, which is problemmatic for long variable-length sentences. While Sutskever et al. [1] addressed that problem by proposing the *source reversing* trick to improve learning, a more elegant approach would be to keep track of a memory of source hidden states and only refer to relevant ones when needed, which is basically the essence of the *attention mechanism* proposed in [11].

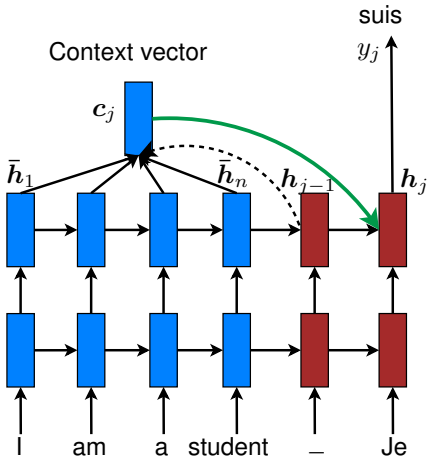


Figure 2: **Attention mechanism** – a simplified view of the attention mechanism proposed in [11]. The attention mechanism involves two steps: first, compute a *context vector* based on the previous hidden state and all the source hidden states; second, use the context vector as an additional information to derive the next hidden state.

Concretely, the attention mechanism will set $\boldsymbol{s} = [\bar{\boldsymbol{h}}_1, \ldots, \bar{\boldsymbol{h}}_n]$ in Eq. (3). The f function now consists of two stages: (a) *attention context* – the previous hidden state $\boldsymbol{h}_{j-1}$ is used to compare with individual source hidden states in $\boldsymbol{s}$ to learn an alignment vector $\boldsymbol{a}_j$; then a context vector $\boldsymbol{c}_j$ is derived as a weighted average of the source hidden states according to $\boldsymbol{a}_j$; and (b) *extended RNN* – the RNN unit is extended to take into account not just the previous hidden state $\boldsymbol{h}_{j-1}$, the current input y_{j-1}, but also the context vector $\boldsymbol{c}_j$ when computing the next hidden state $\boldsymbol{h}_j$. These stages are illustrated in Figure 2.

2.3. Our Models

We follow the attention-based NMT models proposed by Luong et al. [5], which includes two types of attention, *global* and *local*. The *global* model is similar to the one proposed in [11] with some simplifications. The *local* model is, on the other hand, a new model that has a more "focused" attention, i.e., it only puts attention on a subset of source hidden states each time, which results in better performance compared to the global attention approach. We train both types of models so that the ensembling approach as proposed in [1] can benefit from having a variety of models to make decisions.

3. NMT Adaptation

In this section, we explore the possibility of adapting existing models previously trained on one domain to a new domain.

3.1. Training Details

First, we take the existing state-of-the-art English-German system [5], which consists of 8 individual models trained on WMT data with mostly formal texts (4.5M sentence pairs). We then further train on the English-German spoken language data provided by IWSLT 2015 (200K sentence pairs). We use the default Moses tokenizer. The vocabularies are limited to the top 50K frequent words in the WMT data for each language. All other words not in the vocabularies are represented by the special token <unk>. We use the TED tst2012 as a validation dataset for early stopping and report results in BLEU [12] for TED tst2013 (during development) and tst2014, tst2015 (during evaluation).

Our models are deep LSTM networks of 4 layers with 1000-dimensional embeddings and LSTM cells. We further train existing models for 12 epochs in which after the first epoch, learning rates (initially set to 1.0) are halved every two epochs. Effective techniques are applied such as dropout [13], source reversing [1], attention mechanism [11, 5], and rare word handling [3, 4]. More details of these techniques and other hyperparameters can be found in [5]. It takes about 3-5 hours to train a model on a Tesla K40.

3.2. Results

As highlighted in Table 1, adaptation turns out to be very useful for NMT which gives an absolute gain of +3.8 BLEU points compared to using an original model without further training. Additionally, by ensembling multiple models as done in [1], we can achieve another significant gain of +2.0 BLEU points on top of the single adapted model. Compared to the best entry in IWSLT'14 [14], we have advanced the *state-of-the-art* result by +5.2 BLEU points.

System	BLEU
IWSLT'14 best entry [14]	26.2
Our systems	
Single NMT (non-adapted)	25.6
Single NMT (adapted)	29.4 (+3.8)
Ensemble NMT (adapted)	**31.4** (+2.0)

Table 1: *English-German results on TED tst2013* – BLEU scores of various systems. Progressive gains between our systems are given in parentheses.

Furthermore, according to the evaluation results provided by the organizer (Table 2), we are up to +10.0 BLEU points better than the IWSLT'15 baseline system and +4.3 BLEU point better than the best IWSLT'14 entry [14].

System	BLEU	
	tst2014	*tst2015*
IWSLT'14 best entry [14]	23.3	-
IWSLT'15 baseline	18.5	20.1
Our system	**27.6** (+9.1)	**30.1** (+10.0)

Table 2: *English-German evaluation results* – BLEU scores of various systems on the two evaluation sets. We show the differences between our submission and the IWSLT'15 baseline in parentheses.

4. NMT for Low-resource Translation

Until now, state-of-the-art NMT systems rely on large amounts of parallel corpora to sucessfully train translation models such as English-French with 12M-36M sentence pairs [3, 4] and English-German with 4.5M sentence pairs [6, 5]. There is few work examining low-resource translation direction. In [8], the authors examined translation from Turkish to English with 160K sentence pairs, but utilized large monolingual data, the English Gigaword corpus. In this work, we consider applying NMT to the low-resource translation task from English to Vietnamese in IWSLT 2015.

4.1. Training Details

We use the provided English-Vietnamese parallel data (133K sentence pairs). Apart from tokenizing the corpus with the default Moses tokenizer, no other preprocessing step, e.g., lowercasing or running word segmenter for Vietnamese, was done. We preserve casing for words and replace those whose frequencies are less than 5 by <unk>. As a result, our vocabulary sizes are 17K and 7.7K for English and Vietnamese respectively. We use the TED tst2012 as a valid set for early stopping and report BLEU scores on TED tst2013 (during development) and TED tst2015 (during evaluation).

At such a small scale of data, we could not train deep LSTMs with 4 layers as in the English-German case. Instead, we opt for 2-layer LSTM models with 500-dimensional embeddings and LSTM cells. Our other hyperparameters are: (a) we train for 12 epochs using plain SGD; (b) our learning rate is set to 1.0 initially and after 8 epochs, we start to halve the learning rate every epoch; (c) parameters are uniformly initialized in range $[0.1, 0.1]$; (d) gradients are scaled whenever their norms exceed 5; (e) source sentences are reversed which is known to help learning [1], and (f) we use dropout with probability 0.2. We train models with various attention mechanisms, global and local, as detailed in [5]. It takes about 4-7 hours to train a model on a Tesla K40.

4.2. Results

Our results during development are presented in Table 3. Similar to the trend observed in the English-German case, ensembling 9 models significantly boosts the performance by +3.6 BLEU points. Since this is the first time Vietnamese is included in IWSLT, there has not been any published number for us to compare with.

For the final evaluation, our system is, unfortunately, behind the IWSLT baseline as detailed in Table 4. Still, the gap is small and it remains interesting to see how other teams perform. Examining the translation outputs, the first author, as a

System	BLEU
Single NMT	23.3
Ensemble NMT	**26.9**

Table 3: English-Vietnamese results on TED tst2013.

native Vietnamese speaker, was quite amazed at how well the translations can be from an off-the-shelf NMT framework.

System	BLEU
IWSLT'15 baseline	**27.0**
Our system	26.4

Table 4: English-Vietnamese results on TED tst2015 provided by the organizer.

We also notice that the rare word handling technique as often done in NMT [3, 4] yields little gain for our case. We expect that this can be improved by utilizing a Vietnamese word segmenter or simple heuristics to combine collocated words such as the formula used in [15]. The rationale is that many words in English correspond to multiple-character words in Vietnamese such as "success" – "thành công" and "city" – "thành phố". The rare word handling technique requires a word dictionary built from the unsupervised alignments, and in our case, without a segmenter, we are using a word-to-char English-Vietnamese dictionary. As a result, the model will fail when trying to translate English words whose Vietname counterparts are multi-character words.

5. Conclusion

In this work, we have explored the use of Neural Machine Translation (NMT) in the spoken language domain under two interesting scenarios, namely NMT *adaptation* and NMT for *low-resource translation*. We show that NMT adaptation is very effective: models trained on a large amount of data in one domain can be finetuned on a small amount of data in another domain. This boosts the performance of an English-German NMT system by 3.8 BLEU points. This helps advance *state-of-the-art* results in the IWSLT English-German MT track by up to +5.2 BLEU points. For the latter scenario, we demonstrate that an off-the-shelf NMT framework can achieve competitive performance with very little data as in the case of the English to Vietnamese translation direction. For future work, we hope to incorporate phrase-based units in NMT to compensate for the fact that languages like Vietnamese and Chinese often need a word segmenter.

6. Acknowledgment

We gratefully acknowledge support from a gift from Bloomberg L.P. and the support of NVIDIA Corporation with the donation of Tesla K40 GPUs. We thank Thanh-Le Ha for useful discussions and the annonymous reviewers for valuable feedback.

7. References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014.

[2] 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," in *EMNLP*, 2014.

[3] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," in *ACL*, 2015.

[4] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," in *ACL*, 2015.

[5] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP*, 2015.

[6] S. Jean, O. Firat, K. Cho, R. Memisevic, and Y. Bengio, "Montreal neural machine translation systems for WMT'15," in *WMT*, 2015.

[7] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *NAACL*, 2003.

[8] Ç. Gülçehre, O. Firat, K. Xu, K. Cho, L. Barrault, H. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *CoRR*, vol. abs/1503.03535, 2015.

[9] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *EMNLP*, 2013.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[12] K. Papineni, S. Roukos, T. Ward, and W. jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, 2002.

[13] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *CoRR*, vol. abs/1409.2329, 2014.

[14] 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 *IWSLT*, 2014.

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

The English-Vietnamese Machine Translation System for IWSLT 2015

[1,2] Viet Tran Hong, [2,3] Huyen Vu Thuong, [2,4] Trung Le Tien, [2,5]Luan Nghia Pham, [2]Vinh Nguyen Van

[1]University Of Economic And Technical Industries, Hanoi, Vietnam
[2]University of Engineering and Technology-Vietnam National University, Hanoi, Vietnam
[3]ThuyLoi University, Hanoi, Vietnam
[4]Open University, Hanoi, Vietnam
[5]Haiphong University, Haiphong, Vietnam
thviet@uneti.edu.vn,huyenvt@tlu.edu.vn,trunglt@hou.edu.vn,nghialuan@gmail.com,vinhnv@vnu.edu.vn

1. Introduction

In this paper we have described our system for IWSLT2015 machine translation. Focusing primarily on the English-Vietnamese and Vietnamese-English translation direction. Our additions for Moses phrase-based SMT and Phrasal SMT include two language model with monolingual training set for English and Vietnamese.

We submitted two systems to IWSLT 2015 evaluations for English to Vietnamese Machine Translation and Vietnamese to English Machine Translation. Our systems is including sub-systems: 6 based on Phrasal toolkit [Green et al.2014] and 6 others base Moses toolkit [Koehn et al.2007b]. The systems conducted with IWSLT 2015 data using with extension language model using monolingual training data.

2. Data and Pre-Processing

We perform to pre-processing data from IWSLT 2015 for dev, test, train dataset. We convert from formatted xml data to have parallel data. These data are tokenizer for both Vietnamese and English. With Vietnamese data we use VnTokenizer [Phuong-Le Hong2008]. Filter the corrupt characters and the larger sentence of length 300. With English data, we also use tokenizer for segmentation. After that, we conducted experiment for IWSLT 2015 data.

3. Monolingual Data

We expand the language model using Monolingual Data. For English-Vietnam translation, we used data with the crawl from electronic newspaper in Vietnam. We install the tool library used crawler Jsoup to collect 1GB of data and used for training. With Vietnamese-English translation, we use one part of the data WMT2015 collect 1GB of data and used for training.

4. Brief description of the baseline Phrase-based SMT

Phrase-based SMT, as described by [Koehn et al.2003] translates a source sentence into a target sentence by decomposing the source sentence into a sequence of source phrases, which can be any contiguous sequences of words (or tokens treated as words) in the source sentence. For each source phrase, a target phrase translation is selected, and the target phrases are arranged in some order to produce the target sentence. A set of possible translation candidates created in this way is scored according to a weighted linear combination of feature values, and the highest scoring translation candidate is selected as the translation of the source sentence.

Moses [Koehn et al.2007b] is a statistical machine translation system that allows automatically train translation models for any language pair. When we have a trained model, an efficient search algorithm quickly finds the highest probability translation among the exponential number of choices.

Beside Moses, nowadays, Phrasal [Green et al.2014] is also a toolkit for phrase-based SMT. It is a state-of-the-art statistical phrase-based machine translation system, written in Java. At its core, it provides much the same functionality as the core of Moses.

5. Experiment

We present our experiments to translate from English to Vietnamese in a statistical machine translation system. We compare Phrasal and Moses by evaluation with IWSLT 2015 data. We evaluated our approach on English-Vietnamese machine translation tasks, and show that it can significantly outperform the baseline phrase-based SMT system by extended Language model.

The performances of the statistical machine translation systems in our experiments are evaluated by the BLEU scores [Papineni and Zhu2002].

745 sentences in IWSLT15.TED.dev2010 as our dev set on which we tuned the feature weights, and report results on the 1046 sentences of the IWSLT15.TED.tst2015 test set.

Corpus	Sentence pairs	Training Set	Development Set	Test Set
General	123957	122132	745	1080
			English	Vietnamese
Training	Sentences		122132	
	Average Length		15.93	15.58
	Word		1946397	1903504
	Vocabulary		40568	28414
Development	Sentences		745	
	Average Length		16.61	15.97
	Word		12397	11921
	Vocabulary		2230	1986
Test	Sentences		1046	
	Average Length		16.25	16.13
	Word		17023	16889
	Vocabulary		2701	2759

Table 1: The Summary statistical of data sets: English-Vietnamese

In order to extract the translation grammar necessary for our model, we used the provided Europarl and News Commentary parallel training data. The lowercased and tokenized training data was then filtered for length and aligned using the GIZA++ [Och and Ney2003] implementation of IBM Model 4 to obtain one-to-many alignments in both directions and symmetrized by combining both into a single alignment using the grow-diag-final-and method and Berkeley Aligner [DeNero and Klein2007]. We constructed a 4-gram language model using the SRI language modeling toolkit [Stolcke2002] and KenLM [Heafield2011] from the provided English monolingual training data and Vietnamese monolingual training data from crawler web data. Since the beginnings and ends of sentences often display unique characteristics that are not easily captured within the context of the model, and have previously been demonstrated to significantly improve performance, we explicitly annotate beginning and end of sentence markers as part of our translation process. We used the 745 sentences in IWSLT15.TED.dev2010 as our dev set on which we tuned the feature weights, and report results on the 1046 sentences of the IWSLT15.TED.tst2015 test set. (122131 train.tags + 125531 train + 1GB mono data)

6. Evaluation

We conducted some experiments the following:

- Using the state of art Phrase-based SMT Moses:
  - with SMT Moses Decoder [Koehn et al.2007a] and SRILM. We trained a 4 gram language model using interpolate and kndiscount smoothing with 1GB Vietnamese monolingual data for English-Vietnamese translate direction and 1GB English monolingual data for Vietnamese-English translate direction.
  - Before extracting phrase table, we use GIZA++ to build word alignment with grow-diag-final-and algorithm. Besides using pre-processing, we also used default reordering model in Moses Decoder: using word-based extraction (wbe), splitting type of reordering orientation to three class (monotone, swap and discontinuous msd), combining backward and forward direction (bidirectional) and modeling base on both source and target language (fe).
- with SMT Phrasal:
  - We also trained with 1GB Vietnamese monolingual data for English - Vietnamese translate direction and 1GB English monolingual data for Vietnamese-English translate direction a 4 gram language model with 1GB.
  - Before extracting phrase table, we use berkeley aligner to build word alignment with grow-diag-final-and algorithm. Besides using pre-processing, we also used default reordering model in Phrasal.

6.1. English-to-Vietnamese Translation

We conducted 6 experiments: 3 base on Phrasal and 3 base on Moses. Using 4 gram for building language model with monolingual following:

- Using train.tags.en-vi.vi as monolingual data for building language model.
- Combine train.tags.en-vi.vi and train.vi as monolingual data for buiding language model.
- Combine train.tags.en-vi.vi and train.vi and 1GB crawler web data from news site in Vietnam as monolingual data for building language model.

PHRASAL					
No	System	Experiments	BLEU	Description	N-GRAM
1	En-Vn	RUN01	22.16	Baseline System using monoligual data from training set	4
2		RUN02	22.59	Baseline System using monoligual data from 1GB monoligual web crawler data	
3		RUN03	22.90	Baseline System using monoligual data from training set combine training from IWSLT 2015 and 1GB monoligual from web crawler data	
MOSES					
No	System	Experiments	BLEU	Description	N-GRAM
1	En-Vn	RUN01	22.70	Baseline System using monoligual data from training set	4
2		RUN02	22.93	Baseline System using monoligual data from 1GB monoligual web crawler data	
3		RUN03	23.15	Baseline System using monoligual data from training set combine training from IWSLT 2015 and 1GB monoligual from web crawler data	

Figure 1: The experiment our systems for English to Vietnamese translation direction

Figure 1 described results our experiments for English to Vietnamese translation direct. Highest BLEU score is 23.15 for English-Vietnamese translation system with the IWSLT 2015 data.

6.2. Vietnamese-to-English Translation

We conducted 6 experiments: 3 base on Phrasal and 3 base on Moses. Using 4 gram for building language model with monolingual following:

- Using train.tags.vi-en.en as monolingual data for building language model.
- Combine train.tags.vi-en.en and train.en as monolingual data for building language model.
- Combine train.tags.en-vi.en and train.en and 1GB English data from WMT2015 as monolingual data for building language model.

PHRASAL					
No	System	Experiments	BLEU	Description	N-GRAM
1	Vn-En	RUN01	17.37	Baseline System using monoligual data from training set	4
2		RUN02	17.95	Baseline System using monoligual data from 1GB monoligual web crawler data	
3		RUN03	20.18	Baseline System using monoligual data from training set combine training from IWSLT 2015 and 1GB monoligual from web crawler data	
MOSES					
No	System	Experiments	BLEU	Description	N-GRAM
1	Vn-En	RUN01	17.19	Baseline System using monoligual data from training set	4
2		RUN02	17.56	Baseline System using monoligual data from 1GB monoligual web crawler data	
3		RUN03	19.72	Baseline System using monoligual data from training set combine training from IWSLT 2015 and 1GB monoligual from web crawler data	

Figure 2: The experiment our systems for Vietnamese to English translation direction

Figure 2 described results our experiments for Vietnamese to English translation direction. Highest BLEU score is 20.18 for Vietnamese-English translation system with IWSLT 2015 data.

7. Conclusions

In this paper, we has described an an empirical study for English-Vietnamese Statistical Machine Translation. We attempted to tackle the problem of training SMT on parallel data. The extend of the monolingual training set to build language model for training SMT could lead results be more stable and better enough. We evaluated our approach on English-Vietnamese machine translation tasks with Moses toolkit and Phrasal toolkit (state-of-the-art phrase-based and hierarchical statistical MT systems). The experiment results showed that our approach achieved statistically improvements in BLEU scores .

8. Acknowledgements

This work described in this paper has been partially funded by Hanoi National University (QG.15.23 project)

9. References

[DeNero and Klein2007] John DeNero and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 17–24, Prague, Czech Republic, June. Association for Computational Linguistics.

[Green et al.2014] Spence Green, Daniel Cer, and Christopher D. Manning. 2014. Phrasal: A toolkit for new directions in statistical machine translation. In *In Proceddings of the Ninth Workshop on Statistical Machine Translation*.

[Heafield2011] Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom, July.

[Koehn et al.2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pages 127–133. Edmonton, Canada.

[Koehn et al.2007a] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007a. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL, Demonstration Session*.

[Koehn et al.2007b] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007b. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and*

demonstration sessions, pages 177–180. Association for Computational Linguistics.

[Och and Ney2003] Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

[Papineni and Zhu2002] Salim Roukos-Todd Ward Papineni, Kishore and WeiJing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *ACL*.

[Phuong-Le Hong2008] Azim Roussanaly Vinh-Ho Tuong Phuong-Le Hong, Huyen-Nguyen Thi Minh. 2008. A hybrid approach to word segmentation of vietnamese texts. In *In Proceedings of the 2nd International Conference on Language and Automata Theory and Applications*, Springer, LNCS 5196.

[Stolcke2002] Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*, volume 29, pages 901–904.

The IOIT English ASR system for IWSLT 2015

Van Huy Nguyen[1], *Quoc Bao Nguyen*[2], *Tat Thang Vu*[3], *Chi Mai Luong*[3]

[1]Thai Nguyen University of Technology, Vietnam
[2]University of Information and Communication Technology, Thai Nguyen University, Vietnam
[3]Institute of Information and Technology (IOIT),
Vietnamese Academy of Science and Technology, Vietnam

huynguyen@tnut.edu.vn, nqbao@ictu.edu.vn, {vtthang,lcmai}@ioit.ac.vn

Abstract

This paper describes the speech recognition system of IOIT for IWSLT 2015. This year, we focus on improving acoustic and language models by applying some new training techniques based on deep neural networks compared to the last year system. There are two subsystems which are combined by using lattice minimum Bayes-Risk decoding. On the 2013 evaluations set, provided as a test set, we are able to reduce the word error rate of our transcription system from 22.7% of the last year system to 17.6%.

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 talks, in order to interface with the machine translation components in the speech-translation track. The quality of the resulting transcriptions is measured in word error rate (WER).

In this paper, we describe our speech recognition system which participated in the TED ASR track of the IWSLT 2015 evaluation campaign. The system is a further development of our last year's evaluation system [1]. There are two hybrid acoustic models in our system. The first one is built by applying a convolutional deep neural network with the input feature of log Mel filter bank feature (FBANK). The second one is applied a feed-forward deep neural network. Its input feature is a speaker-dependent feature that is extracted by applying a feature space maximum likelihood linear regression (fMLLR) in the speaker adaptive training (SAT) stage of the baseline system. These models and an interpolated language model are used to produce decoding latices which are then used to generate the N-best lists for re-scoring.

The organization of the paper is as follows. Section 2 describes the data that our system is trained on. This is followed by Section 3 which provides a description of the way to extract acoustic features. An overview of the techniques, used to build our acoustic models, is given in Section 4. Language model and dictionary are presented in Section 5. We describe the decoding procedure and results in Section 6 and conclude the paper in Section 7.

2. Training Corpus

For training acoustic models, we used two types of corpus as described in Table 1. The first corpus is TED talk lectures (http://www.ted.com/talks). Approximately 220 hours of audio, distributed among 920 talks, were crawled with their subtitles, which are 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 spoken words, which lead to the necessity for long-speech alignment. Segmenting the TED data into sentence-like units, used for building a training set, is performed with the help of SailAlign tool [2] which helps us to not only acquire the transcript with exact timing, but also to filter non-spoken sounds such as music or applause. A part of these noises are kept for training noise models while most of them are abolished. After that, the remained audio used for training consists of around 160 hours of speech. The second corpus is Libri360 which is the Train-clean-360 subset of the LibriSpeech corpus [3]. It contains 360 hours of speech sampled at 16 kHz, and is available for training and evaluating speech recognition system.

Table 1: Traning data for acoustic models

Corpus	Type	Hours	Speakers	Utts
Ted	Lecture	160	718	107405
Libri360	Audiobook	360	921	104014

3. Feature Extraction

In this work, two kinds of acoustic feature are used for developing the acoustic models. The first one is a Mel-frequency Cepstral Coefficients (MFCC). A Hamming window of 25ms, which is shifted at the interval of 10ms, is applied. Each MFCC vector consists of 39 coefficients which are 13 MFCCs, the first and the second order derivatives. The second kind is a combination of a log Mel filter bank feature and a pitch feature (FBANK+P). FBANK+P consists of 43 coefficients including 40 FBANK coefficients, 1 the pitch value, the first derivative of the pitch value, and the probability of voice for the current frame. Both MFCC and FBANK+P are extracted by using the Kaldi toolkit [4][5].

4. Acoustic Model

4.1. Baseline Acoustic Model

The baseline acoustic model was built by using the Kaldi toolkit [4] with MFCC feature. First, this model was trained as a basic context dependent tri-phone model, followed by a speaker adaptive training (SAT) with a feature space maximum likelihood linear regression (fMLLR). A discriminative training based on the maximum mutual information (MMI) was applied at the end. This model (MMI-SAT/HMM-GMM) had 6496 tri-phone tied states with 160180 Gaussian components, and it was then used to produce a forced alignment in order to get the labeled data for training deep neural networks.

4.2. Hybrid Acoustic Model

The hybrid Deep Neural Network and Hidden Markov Model (DNN-HMM) acoustic model were built in which the HMM models were the baseline model's HMM, and their deep neural networks were built in different architectures. Fig. 1 describes the process for training these models. The first hybrid model was applied a feedforward deep neural network (DNN) congured as 440-1024*5-6496 (input layer with 440 neurons, 5 hidden layers with 1024 neurons for each, output layer with 6496 neurons). The second one was applied a convolution neural network (CNN-DNN) which has one convolutional layer with convolution and polling operations. The conguration of the convolutional layer was as follows: 128 filters with filter size and shift as 9 and 1 for each. The pooling width and shift is set to 2 and 2, respectively. The output from the pooling layer was further processed with feedforward DNN with 5 hidden layers (1024 neurons each), and output layer with 6496 neurons. For training DNN and CNN-DNN, 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) [6] was adopted for the second stage training. The input feature for the DNN was a fMLLR-based feature that was calculated as follow: The MFCC was adjusted by concatenating 11 neighbor vectors (5 ones for each left and right side of the current MFCC vector) to make the context dependent feature, afterward the dimension of the concatenated vector was reduced to 40 by applying a linear discriminate analysis (LDA) and decorrelated with a maximum likelihood linear transformation (MLLT). It is finally applied a feature space maximum likelihood linear regression (fMLLR) in the speaker adaptive training (SAT) stage. The LDA, MLLT and fMLLR transforms are estimated during the training of the baseline model. The concatenation of 11 neighbor vectors of FBANK+P, the first and the second order derivatives was used as input feature of CNN-DNN.

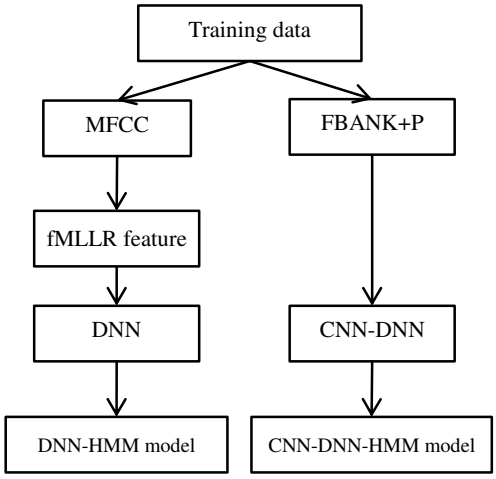


Figure 1: Training process of hybrid acoustic models

5. Language Model and Dictionary

Two categories of textual corpora was used for estimating the language model (LM) (as shown in Table 2). The first one is the transcript of Libri360 data set that was used for training the acoustic models. The second one is the subtitles of all TED talks published before June-2015 (TED2015) which is provided by Fondazione Bruno Kessler (FBK) (https://wit3.fbk.eu). TED2015 was used for training the language model after rejecting all disallowed TED talks according to the suggestion of IWSLT-2015 committee.

Table 2: Traning data for language model

Corpus	Utts
Libri360	104014
TED2015	517098

For training the language model, a vocabulary set is firstly extracted from textual sets. This vocabulary set has 73491 words and is then used to build the language model by using the SRILM toolkit [7]. The perplexity (PPL) score of the trained language model is 184 on the tst2013 test set. In order to improve the performance, it is then combined in weight of 0.65 with a 3-gram Gigaword Language model that

is available on [8] by using the linear interpolation method. We implemented combinations with difference weights from 0.1 to 0.9 (step is 0.5). The weight of 0.65 is the weight that gave a minimum PPL of 151 on tst2013.

The vocabulary set, obtained in the training stage of the language model, is used to make the dictionary. The lexicon is 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.

6. Decoding Procedure and Results

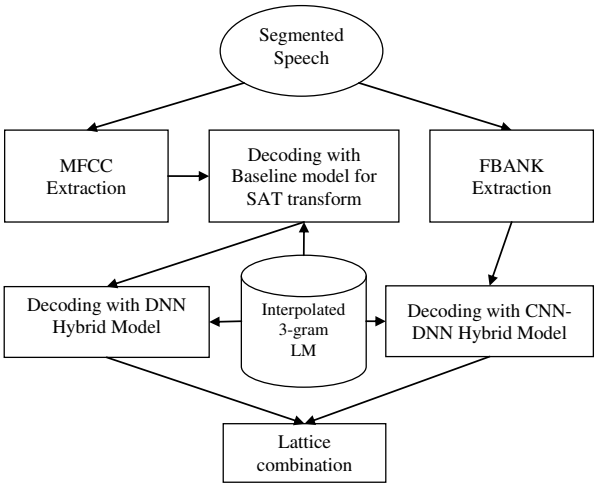


Figure 2: The full decoder architecture

During development, we evaluated our system on the tst2013 test set that released by the IWSLT organizers. Fig. 2 shows our complete decoding process. After feature extraction step, followed by decoding with the baseline system to estimate the transforms LDA, MLLT, and fMLLR, we operated two parallel decoding sequences for the hybrid acoustic models. For each model, the complete process consists of a decoding with the 3-gram LM applying Kaldi decoder. Lattice outputs from the this pass are combined by using Lattice Minimum Bayes-Risk (MBR) decoding as described in [10].

Table 3 lists the performance of our system in terms of the word error rate (WER). Both tst2013 and dev2012 sets were segmented manually. Regarding the performance of the baseline system, the WER is 18.53% on dev2012 and 22.86% on tst2013. The first row is the number of the best system from last year [1] on the same test set. As we can see on the Table, all of our hybrid models give better WERs which are approximately 3% absolute compared to the baseline model. The last row on the table shows the final combination results of the hybrid models that give a further 1% absolute WER reduction as compared to the best single system. For this year's test set which was segmented automatically like last year system [1], we obtained 14.4% WER (about 2 % loss compared to manual segmentation).

Table 3: Experiment results

Denoted	**Model**	**WER%**	
		dev2012	**tst2013**
Last year	Combination	18.7	22.7
Baseline	MMI-SAT/HMM-GMM	18.53	22.86
S1	DNN-HMM	15.19	18.85
S2	CNN-DNN-HMM	15.81	19.30
S1+S2	Combination	14.5	17.6

7. Conclusions

In this paper, we presented our English LVCSR system, with which we participated in the 2015 IWSLT evaluation. The acoustic model was improved by using deep neural networks for this year evaluation. On the 2012 development set for the IWSLT lecture task our system achieved a WER of 14.5%, and a WER of 17.6% on the 2013 test set. The final combination model gives about 5% absolute WER reduction on tst2013 compared to the last year system.

In the future, we intend to improve language model using deep neural network as in [11] as well as will apply a hybrid DNN on top of deep bottleneck features [12] to improve acoustic model for the systems.

8. Acknowledgements

This work is partially supported by Project: "Development of spoken electronics newspaper system based on Vietnamese text-to-speech and web-based technology", VAST01.02/14-15

9. References

[1] 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 for Spoken Language Translation (IWSLT)*, Lake Tahoe, USA, Dec-2014 2014.

[2] 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.

[3] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP)*. South Brisbane: IEEE, 2015, pp. 5206 – 5210.

[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. 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.

[5] P. Ghahremani and D. . R. K. . T. J. . K. S. BabaAli, B. ; Povey, "A pitch extraction algorithm tuned for automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 2494 – 2498.

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

[7] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *International Symposium on Chinese Spoken Language Processing (ISCSLP)*, Hong Kong, 2012.

[8] K. Vertanen, *English Gigaword language model training recipe*, Std. [Online]. Available: https://www.keithv.com/software/giga/

[9] S. Meignier and T. Merlin, "Lium spkdiarization: an open source toolkit for diarization," in *in CMU SPUD Workshop*, 2010.

[10] 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.

[11] 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.

[12] 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.

The I^{2}R ASR System for IWSLT 2015

Tran Huy Dat, Jonathan William Dennis, Ng Wen Zheng Terence

Human Language Technology Department
Institute for Infocomm Research, A*STAR, Singapore
{hdtran,jonathan-dennis,wztng}@i2r.a-star.edu.sg

Abstract

In this paper, we introduce the system developed at the Institute for Infocomm Research (I^{2}R) for the English ASR task within the IWSLT 2015 evaluation campaign. The front-end module of our system includes a harmonic modelling based automatic segmentation and the conventional MFCC feature extraction. The back-end module consists of an auxiliary GMM-HMM training to provide the speaker adaptive transform (SAT) and the initial forced alignment, followed by a discriminative training DNN acoustic modelling. Multi-stage decoding strategy is employed with a semi-supervised DNN adaptation which uses weighted labels generated by the previous-pass decoding output to update the trained DNN models. Finally, Recurrent Neural network (RNN) is used to train and rescore the language modelling to further improve the performances. Our system achieved 8.4 % WER on the tst2013 development set, which is better than the official results on the same set reported from the previous evaluation. For this year's tst2015 test set, we obtained 7.7% WER.

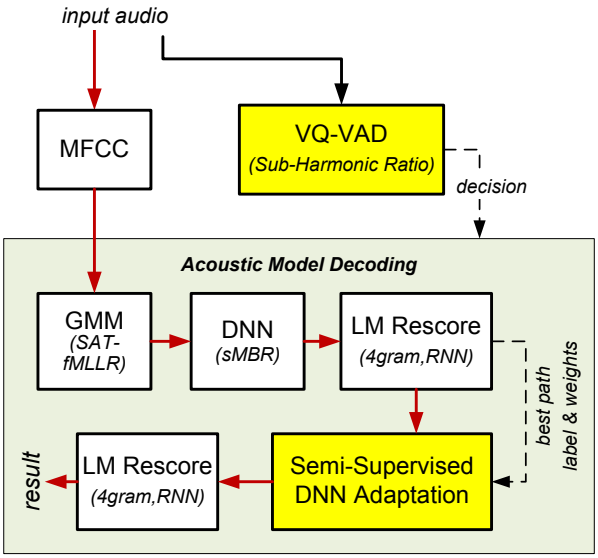


Figure 1: Overview of the I^{2}R ASR system for IWSLT 2015.

1. Introduction

The goal of the Automatic Speech Recognition (ASR) track for IWSLT 2015 is to transcribe TED talks and TEDx talks [1]. The speech in English TED talks are lectures related to Technology, Entertainment and Design (TED) in spontaneous speaking style. Despite that the speech in the TED talks is in general planned, well articulated, and recorded in high quality, the task is challenging due to the large variability of topics, the presence of non-speech events, the ascents of non-native speakers, and the informal speaking style. In this paper, we introduce our system for English TED ASR track of the 2015 IWSLT evaluation campaign. We choose to focus on developing a single system rather than a fusion of multiple platforms. The overview of our ASR system is illustrated in Fig.1. Since the TEDs' audio samples, during the test phase, are provided without class labels and timing information, automatic segmentation is necessary to split audio file into speech sentences to input the ASR system. In this work, we develop a voice activity detection (VAD) method based on harmonic modelling of speech signals and build the automatic segmentation on top of that. As the TEDs audio is normally recorded in relatively high quality, no noise compensation method is needed and we just apply the conventional MFCC features as the input to the ASR system. The training is started with an auxiliary GMM-HMM training to provide the speaker adaptive transform (SAT) and the initial alignment. Then the DNN acoustic modelling is carried out on top of SAT features with a fixed size concatenating window. The hidden layer weights are initialised using layer-wise restricted Boltzmann machine (RBM) pre-training, using 100 hours of randomly selected utterances from the training materials. Multi-stage decoding strategy is employed with semi-supervised DNN model adaptation using weighted lattices generated by the previous-pass decoding output. Finally, Recurrent Neural network (RNN) is used to train and re-score the language modelling to further improve the performances. Our system obtained WER of 8.4% on the development set (tst2013) and 7.7% on the test set (tst2015), respectively. The organisation of the rest of the paper is as follows. Secs.2 introduces the automatic segmentation. Secs.3 and 4 describes the acoustic modelling and language modelling, respectively. Secs.5 reports the experimental results and analyzes the role of each module into the ASR performances. Finally, Secs 6 concludes the paper.

2. Automatic Segmentation

The VAD module detects the speech segments based on the harmonic to sub-harmonic ratio, and uses an adaptive threshold to reject regions of noise and other non-speech and a post-processing to smooth the result.

Our approach uses a vector quantisation (VQ) system as the basis for voice activity detection (VAD), with frame selection based on both energy and the harmonic to sub-harmonic ratio (SHR) [2, 3], which is a feature for voiced speech detection. Three acoustic categories are targeted in this knowledge-based approach:

Speech - voiced speech is characterised by having both a high SHR and high energy, due to the strong harmonic structure produced during speech vocalisation.

Background Noise - for the task of lecture-style speech, where the signal-to-noise ratio (SNR) is high, the noise will typically have a much lower energy than the speech signal.

Clapping - impulsive noise has a high energy but a low SHR, which is due to the physical nature of the way the sounds are generated.

To compute the SHR within each short-time windowed frame, using a frame length of 32 ms, the amplitude spectrum $E(f)$ is first computed. For voiced segments of speech, $E(f)$ has strong peaks at the harmonics of the fundamental frequency $F0$. From this spectrum, the summation of harmonic amplitude (SHA) and summation of sub-harmonic amplitude (SSA) is computed for each frequency in the range $[F0_{min}, F0_{max}]$ as follows:

$$SHA(f) = \sum_{k=1}^{N_{harm}} \sum_{a=-\Delta}^{\Delta} E(k.f + a) \quad (1)$$

$$SSA(f) = \sum_{k=1}^{N_{harm}} \sum_{a=-\Delta}^{\Delta} E((k - \frac{1}{2}).f + a) \quad (2)$$

where only the first N_{harm} harmonics are taken into account in the summation, and a window of $\Delta = 1$ neighbouring bins are included in the summation to account for inharmonicity. Finally, the harmonic to sub-harmonic ratio (SHR) is the ratio of the two, as follows:

$$SHR(f) = \frac{SHA(f)}{SSA(f)} \quad (3)$$

where the maximum value $\max_f (SHR(f))$ is taken as the value of the feature for each frame, $SHR[t]$.

The VQ process is applied on each TED talk independently, and uses basic Mel-frequency cepstral coefficient (MFCC) as the underlying features. Our approach is to use k-Means clustering to build a set of representative vectors for each of the three categories. The top 10% of the available frames, ranked according to the above-mentioned frame-selection criteria, are used for both the speech and noise categories, while only the top 2% of frames are used for the clapping category in anticipation that less data is available.

To allow a threshold to be set for the VAD, the VQ distances are compared using the following formula:

$$VQR = \min(D_{noise}, D_{clapping}) - min(D_{speech}) \quad (4)$$

where the distances D for each category are calculated as the minimum Euclidean distance of the quantised vectors for that category. We used a threshold set at $thresh = 0$ such that speech frames are those with $VQR > thresh$.

Note that the frame-level output decision is first smoothed to join together segments separated by a gap of less than 500 milliseconds, with an additional hangover of length 500 milliseconds then applied to ensure that unvoiced speech at the start and end of the segments are not missed.

3. Acoustic Modelling

This section describes the acoustic modelling used in the I^{2}R ASR system, as shown in Figure 1. The following three aspects are detailed: (1) training data selection, (2) feature extraction and auxiliary GMM-HMM, and (3) DNN acoustic modelling.

3.1. Training Data

Following the success of the NICT system for IWSLT 2014 [4], we use a similar set of training data based on the following three corpora:

Wall Street Journal - this comprises of 81.1 hours of read speech, available from the Linguistic Data Consortium (LDC), from LDC93S6B and LDC94S13B.

HUB4 English Broadcast news - unlike [4] we use the full 201 hours of broadcast news data from LDC97S44 and LDC98S71.

TEDLIUM version 2 - this corpus contains 204 hours of lecture-style TED speech [5] consisting of 1481 talks after the removal of non-permissible talks.

Further experiments were conducted with an additional 44 hours of data extracted from the Euronews corpus [6], provided by the organisers. However, this was found to degrade the WER results by approximately 4% relative so in the final system we did not include it in the training.

3.2. Feature Extraction and Auxiliary GMM-HMM

The acoustic models (both GMM-HMM and DNN) are trained on 13-dimensional MFCCs, without energy, which are mean normalised over the speech segments extracted from each conversation for the speaker. Later, these features are spliced by ± 3 frames adjacent to the central frame and

projected down to 40 dimensions using linear discriminant analysis (LDA).

Prior to DNN training, an auxiliary GMM-HMM is first trained to provide speaker adaptive transforms (SAT) and the initial alignments for training the subsequent DNN system by forced alignment, which inherits the same tied-state structure. To train the GMM-HMM, a monophone system is first trained using the shortest twenty thousand utterances, to make the initial alignments based on a flat-start approach easier. Next, triphone and LDA GMM-HMM systems are trained with 2500 and 4000 tied states respectively, followed by SAT training to give a final SAT GMM-HMM system with 6353 tied triphone states and 150k Gaussians. The SAT approach uses feature-space maximum likelihood linear regression (fMLLR) transforms, with speech segments extracted from each conversation assumed to come from the same speaker. For training, the fMLLR transforms are computed from forced alignments, while for testing, the fMLLR transforms are computed from lattices by using 2 passes of decoding.

3.3. DNN Acoustic Modelling

The DNN acoustic model is trained on top of SAT features that are spliced ± 5 frames and rescaled to have zero mean and unit variance. The DNN has 5 hidden layers, where each hidden layer has 2048 sigmoid neurons, and a 6353 dimensional softmax output layer. The hidden layer weights are initialised using layer-wise restricted Boltzmann machine (RBM) pretraining, using 100 hours of randomly selected utterances from the TEDLIUM corpus [5]. After pretraining, fine-tuning is performed to minimize the per-frame cross-entropy between the labels and network output. The first stage of fine-tuning was performed using the same 100 hour subset as for pretraining with a learning rate of 0.008 and halving beginning when the network improvement slows. This then generated alignments for a full training set to perform a second stage of fine-tuning. Finally, the DNN is retrained by sequence-discriminative training to optimise the state minimum Bayes risk (sMBR) objective. Two iterations are performed with a fixed learning rate of 1e-5. The Kaldi toolkit is used for all experiments [7].

3.4. Semi-supervised DNN adaptation

During decoding, semi-supervised DNN adaptation is utilised on a per-talk basis to reduce any mismatch between training and testing conditions and to provide speaker adaptation of the acoustic model [8, 9]. Additional iterations of fine-training of the DNN requires a frame-level label, and potentially also a confidence measure, and these are generated based on the initial output of the system, as shown in Figure 1.

The frame-level confidence c_{frame_i} is extracted from the lattice posteriors $\gamma(i, s)$, which express the probability of being in state s at time i. The decoding output gives us the best path state sequence, $s_{i,1best}$, and the confidence values are the posteriors under this sequence, as follows [9]:

$$c_{frame_i} = \gamma\left(i, s_{i,1best}\right) \tag{5}$$

The best path state sequence and confidence measures are then used as the target labels and weightings respectively for additional iterations of DNN fine-tuning, with weights less than $c = 0.7$ set to zero. In our experiments, all weights in the network are updated, as our experiments suggested this performed better than adapting only the first layer of the DNN. The learning rate is 0.0008, with halving performed each iteration until no improvement is observed.

Category	Corpus	Sentences selected	Pct% of Original
In-domain	TED Talks	92k	-
Out-of-domain	CommonCrawl	770k	9%
	Europarl	140k	6%
	Gigaword FR-EN	0.9M	4%
	NewsCommentary	47k	19%
	News	12.3M	18%
	Yandex	310k	31%

Table 1: Training data for the language models.

4. Language Modelling

This section describes the language modelling and rescoring approaches used in the I^{2}R ASR system. The following three aspects are detailed: (1) training data selection, (2) n-gram language model training, and (3) RNN language modelling and rescoring.

4.1. Training Data and N-gram Language Model

Table 1 shows the data used for training the language models in the I^{2}R ASR system. The out-of-domain data is provided as part of the enhanced TEDLIUM version 2 corpus [5], and consists of text selected from corpus from the WMT 2013 evaluation campaign. The selection is based on the XenC tool [10], which is a filtering framework that trains both in-domain and out-of-domain language models and uses the difference in the computed scores on the out-of-domain text as an estimation of the closeness of those sentences to the in-domain subject. Text from each corpus is concatenated together to form a single large set that is used for training each of the subsequent language models.

Two n-gram language models are trained using the data selected from the available corpus as described above. The first is a 3-gram model, trained using the "Kaldi LM" package [7], which is used for DNN-based lattice generation during the first pass of decoding. The second is a 4-gram model, which is trained in an identical fashion to the one above, and is used for rescoring of the word lattice to provide a consistent improvement in WER performance.

Processing Step	WER tst2013	
	Ground Truth	Segmentation
SAT GMM	21.3%	22.4%
DNN sMBR	12.3%	11.6%
+ LM Rescore	10.8%	10.1%
+ DNN Adapt 1	9.5%	8.7%
+ DNN Adapt 2	9.4%	8.5%
+ DNN Adapt 3	9.1%	8.4%

Table 2: Detailed experimental results on the tst2013 development set showing the performance at each stage of the decoding system. Note that the DNN semi-supervised adaptation step includes a final round of language model rescoring.

4.2. RNN Language Model Training and Rescoring

A recurrent neural network (RNN) language model is trained and used for n-best list rescoring to further enhance the WER performance. The RNNLM package version 0.3e [11] was used, with 30k words in the vocabulary, 480 hidden units, 300 classes, and 2000M direct connections. Back-propagation Through Time (BPTT), with truncated time order 5, was used for RNN training, which performs joint training with a maximum entropy model to reduce the hidden layer size. The training data for the RNN was the same as above, although to enable a faster training time a random subset of 2M sentences (14% of the filtered corpora) were selected for training.

The RNN language model has a perplexity of approximately 60, and is used to rescore the output decoding lattice, with interpolation weight of 0.3 instead of using the 4-gram LM. With lower perplexity, the RNN language model can be beneficial in reducing the WER, since final ASR performance is quite dependent on a strong language model. Note that the CMU pronouncing dictionary [12] was used, limited to the words that appear in the language training databases.

5. Experimental Results

In this paper, we opt to use a single system without any combination using ROVER [13] or other techniques. At the decoding stage, we first decode the whole test set from the trained DNN acoustic models and 3-gram LM. Then the 4-gram LM rescoring is carried out, following by another RNN rescoring, described above. Next, the semi-supervised adaptation is applied for each TED test file. Each round of semi-supervised adaptation includes DNN models lattice outputting, 4-gram LM rescoring, RNN LM rescoring and DNN model adaptation. After 3-rounds of semi-supervised adaptation of the DNN acoustic model, there was no further improvement in WER on the devleopment sets, hence we applied the same number during final testing. For this year's tst2015 test set, we obtained 7.7% WER.

Processing Step	WER Gain (tst2013)
DNN sMBR	9%
+ LM Rescoring	1.5%
+ Semi-supervised DNN	1.7%

Table 3: Comparison of the approximate WER improvements given by the key components of the system, compared to the SAT-GMM result.

5.1. Results and Discussions

Table 2 reports detailed experimental results on the tst2013 development set showing the performance at each stage of the training and decoding with ground truth segmentation and the proposed automatic segmentation. We can see that the performance of the proposed segmentation is comparable to the ground truths at the baseline SAT-GMM models and and even outperformed the latter at the more comprehensive training models. The best result from tst2013 development set is 8.4% WER and it was obtained with multi-stage semi-supervised adaptation with rescoring of LM. This result is better than the official result of 10.6% WER on the same tst2013 set from last evaluation. The DNN with sMBR discriminative training yields a reasonable result of 11.6% WER and that system is fast enough to be real-time and hence recommended for the live engines.

5.2. Analysis of Word Error Rate Improvements

A summary of the contribution of each processing step to the final WER result is shown in Table 3. It can be seen that the DNN with sMBR discriminative training gives the most significant improvement in performance over the baseline SAT-GMM. In addition, the DNN decoding strategy gives a total of around 2-3% improvement, with the biggest contribution coming from the semi-supervised DNN speaker adaptation, combined with a consistent improvement achieved through language model rescoring. The semi-supervised DNN adaptation is suitable for TED and TEDx talks since it involves a single speaker and long enough to be effective. However, a big jump of performance is normally seen in the first round of adaptation while it is very time consuming. Hence, in practical situations, using one round of adaptation is recommended.

6. Conclusions

In this paper, we described our English ASR system for IWSLT 2015 evaluation campaign. This is a single system consisting of harmonic modelling voice activity detection (VAD) for automatic segmentation, speaker adaptive training (SAT) GMM-HMM initial forced alignment, DNN acoustic modelling with sMBR discriminative training, RNN language modelling and rescoring, and semi-supervised DNN adaptation in decoding. We obtained good performances on both the development and test sets. Among the system, the

harmonic modelling VAD, the DNN acoustic modelling with discriminative training, the semi-supervised DNN adaptation have found to be the key components which contributed to the ASR improvements compared to the baseline systems.

7. References

[1] "Ted," https://www.ted.com/talks.

[2] X. Sun, "Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1. IEEE, 2002, pp. I–333.

[3] T. Drugman and A. Alwan, "Joint robust voicing detection and pitch estimation based on residual harmonics." in *Interspeech*, 2011, pp. 1973–1976.

[4] P. Shen, X. Lu, X. Hu, N. Kanda, M. Saiko, and C. Hori, "The NICT ASR system for IWSLT 2014," in *Proceedings of IWSLT 2014*, 2014, pp. 113–118.

[5] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks," in *Proc. of LREC*, 2014, pp. 3935–3939.

[6] R. Gretter, ""Euronews: a multilingual speech corpus for ASR," in *Proceedings of LREC*, 2012, pp. 4161–4164.

[7] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, Ondej Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The kaldi speech recognition toolkit," in *IEEE workshop on automatic speech recognition and understanding (ASRU)*. IEEE, 2011.

[8] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7947–7951.

[9] K. Vesely, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 267–272.

[10] 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.

[11] T. Mikolov, "Statistical language models based on neural networks," 2012.

[12] C. M. University, "The carnegie mellon university pronouncing dictionary v07a," in *[Online] http://www.speech.cs.cmu.edu/cgi-bin/cmudict*, 2015.

[13] J. Fiscus, "A post-processing system to yield reduced error rates: recognizer output voting error reduction (ROVER)," in *Proceedings of LREC*, 1997, p. 347354.

The JAIST-UET-MITI Machine Translation Systems for IWSLT 2015

Hai-Long Trieu[1], *Thanh-Quyen Dang*[2], *Phuong-Thai Nguyen*[2], *Le-Minh Nguyen*[1]

[1]Japan Advanced Institute of Science and Technology
[2]University of Engineering and Technology, Vietnam National University, Hanoi
trieulh@jaist.ac.jp, dangthanhquyen@gmail.com, thainp@vnu.edu.vn, nguyenml@jaist.ac.jp

Abstract

This paper describes the submission of the Japan Advanced Institute of Science and Technology and the University of Engineering and Technology, Vietnam National University, Hanoi for the machine translation track of the IWSLT 2015 workshop. We participated in the shared task for the language pair: English-Vietnamese. First, we investigate and apply some approaches and techniques including phrase-based, syntax-based and domain adaptation for the TED talks domain. Then, we observe and evaluate experimental results of these systems on the development sets to setup the best configurations. Experimental results show that the phrase-based systems obtain the best performance on this domain in comparison with the other applied approaches.

Keywords: phrase-based machine translation, syntax-based machine translation, domain adaptation

1. Introduction

This year's machine translation track of the IWSLT workshop is for language pairs: English paired with French, German, Chinese, Czech, Thai, and Vietnamese. We participate in both translation directions for English-Vietnamese.

We approach the task by first investigating some effective existing methods: phrase-based and syntax-based. Phrase-based translation systems (Koehn et al., 2003 [16], Chiang, 2007 [5]) achieve state-of-the-art results in many typologically diverse language pairs. For this shared task, we participate in translation for English-Vietnamese, a diverse language pair with many different characteristics in linguistic; therefore, we try to apply the syntax-based approach to exploit linguistic knowledge. For the phrase-based methods, we built our systems based on the Moses toolkit (Koehn et al., 2007 [15]). For the syntax-based methods, we applied the open source Joshua [17] with two particular SCFG types: Hiero [5] and Syntax Augmented Machine Translation (SAMT) [34]. In addition to these two methods, because we used unconstrained data in training our models, we conducted experiments on some domain adaptation techniques including: fill-up [2] and back-off [1] to leverage more improvements in our systems. We evaluated our systems on tuning data sets provided by the workshop.

The rest of this paper is organized as follows: in Section 2, we discuss some linguistic characteristics of the diverse language pair English-Vietnamese and review some previous researches of machine translation for English-Vietnamese. In Section 3, we describe a general system overview with details on our training pipeline and decoder configuration. Next we present empirical results for the individual translation directions. In Section 5, we investigate some challenges in the translation task for TED data. We analyze translation errors and experimental results in Section 6. Finally, conclusions are described in Section 7.

2. English-Vietnamese Machine Translation

In this section, we discuss different characteristics between English and Vietnamese. Then, we review some previous researches related to English-Vietnamese machine translation.

2.1. English vs. Vietnamese: Some Linguistic Characteristics

There are many different characteristics between English and Vietnamese languages. For instance, in word order, adjectives follow nouns in Vietnamese while this order is converse in Vietnamese. In another aspect, English uses morphological morphemes to mark tense and number, whereas Vietnamese uses words that precede the verb to mark tense and the addition of numerals and quantifiers for indicating numbers. See Table 5 of [30] for more details of these comparisons.

2.2. Previous Work

Dinh et al., 2003 [8] presented a hybrid model for machine translation (MT) which combines rule-based MT and corpus-based MT (Bitext-Transfer Learning) that learns from bilingual corpus to extract disambiguating rules. Rule-based MT systems were improved by using word-order transfer [9]. This model has been experimented in English-to-Vietnamese MT system (EVT). Ho et al., 2008 [1] built an English-Vietnamese statistical machine translation (SMT) system namely EVSMT1.0 based on the framework of the open

[1]http://www.statmt.org/moses/?n=Advanced.Domain#ntoc3

source Moses and showed potential features in comparison with an existing commercial MT using traditional rule-based approach.

For experiments on the language pair English-Vietnamese, Nguyen and Shimazu 2006 [22] proposed a syntactic transformation model in the pre-processing phase which reorder the structure of source sentence so that it is closer to the structure of target sentence. The transformation is also produced by a dependency-based parser together with a set of hand-crafted rules [13]. Nguyen et al., 2006 used linguistic knowledge of languages in the preprocessing phase using a morphological analysis or POS tagger on the source sentence [21]. Nguyen et al., 2008 [32] proposed reordering at trunk level and incorporate the global reordering model into the decoder. Related to syntactic approaches, Nguyen et al., 2008 [23] applied a tree-to-string phrase-based method which employs a syntax-based reordering model in the decoding phase.

There have been efforts in developing English-Vietnamese bilingual corpora. Nguyen et al., 2006 [31] described dictionaries used in English-Vietnamese Machine Translation (EVMT). Another work of building bilingual corpus was conducted in the National project VLSP (Vietnamese Language and Speech Processing). [2] In this project, an English-Vietnamese bilingual corpus was built, which includes more than 100,000 sentence pairs. English-Vietnamese corpora were also built at different levels including a study on building POS-tagger for bilingual corpora or building a bilingual corpus for word sense disambiguation ([6], [7]). This task was also shown in some other researches ([18], [19], [20]).

3. System Overview

3.1. Pre-processing

We pre-processed English training data by using scripts from the Moses toolkit including tokenization, and then truecasing. For Vietnamese training data, we used JVnTextPro[3] for tokenization. We remove sentences longer than 80 words and their corresponding translations.

3.2. Word Alignment

Word alignment was computed using the first three steps of the train-factored-phrasemodel.perl script packed with Moses (Koehn et al., 2007). We used MGIZA++ (Gao and Vogel, 2008) [11], a multi-threaded implementation of GIZA++ (Och and Ney, 2003) [25] using the *grow-diag-final-and* heuristic (Koehn et al., 2003) [16].

3.3. Language Model

We used all available monolingual data and KenLM [12] to train interpolated Kneser-Ney discounted 5-gram LMs for each system.

3.4. Baseline Features

We follow the standard approach to SMT of scoring translation hypotheses using a weighted linear combination of features. The core features of our models are a 5-gram LM score, phrase translation and lexical translation scores, word and phrase penalties, and a linear distortion score.

We used the hierarchical lexicalized reordering model (Galley and Manning, 2008) [10] with 4 possible orientations (monotone, swap, discontinuous left and discontinuous right) in both left-to-right and right-to-left direction with the setup *msd-bidirectional-fe lexicalized* reordering.

3.5. Tuning and Decoding

The feature weights were tuned using k-best batch MIRA (Cherry and Foster, 2012) [4]. This is a version of MIRA (a margin based classification algorithm) which works within a batch tuning framework. We set the number of inner MIRA loops to 300 passes over the data.

4. Experimental Results

In this section we describe peculiarities of individual systems and present experimental results.

4.1. Data

4.1.1. Bilingual Data

In addition to the data provided by the workshop [4] (constrained data) [3], we used unconstrained data including bilingual corpora for training translation models and monolingual corpora for training language model (LM). Bilingual corpora and sentence length statistics are indicated in Table 1 and Table 2.

The unconstrained bilingual data include several resources in which we used the English-Vietnamese bilingual corpus provided by the National project VLSP (Vietnamese Language and Speech Processing).[5] This corpus includes 80,000 sentence pairs in Economics-Social topics and 20,000 sentence pairs in information technology topic. In addition, we used the EVBCorpus including texts extracted from books, fictions or short stories, law documents, and newspaper articles and then translated by skilled translators [19], [20]. We also used our in-house data including bilingual sentences extracted from newspaper articles. We combines these datasets and obtained 419,385 unconstrained parallel sentences.

For development data, we experimented and evaluated our systems on various tuning sets: each particular set of five development sets (*dev2010*, *tst2010*, *tst2011*, *tst2012*, *tst2013*) provided by the workshop and a set of merging all

[2] http://vlsp.vietlp.org:8080/demo/?page=home
[3] http://jvntextpro.sourceforge.net/
[4] https://wit3.fbk.eu/mt.php?release=2015-01
[5] http://vlsp.vietlp.org:8080/demo/?page=home

these five sets. We setup the development set *tst2013* which shows the best performance for tuning data.

Table 1: Bilingual Corpora. Language codes: en=English, vi=Vietnamese.

Corpus	SentPairs	Tokens en	Tokens vi
Constrained	133,082	54,139	26,867
Unconstrained	419,385	84,506	41,120
Development	1,304	3,918	2,694

Table 2: Sentence Length Statistics. Len.Avg: average sentence length on the corpus. Len.Max: the maximum sentence length. Len>80: number of sentences which length >80.

Corpus	Len.Avg	Len.Max	Len>80
train.en	17.33	513	341
train.vi	22.66	735	1572
test.en	16.54	90	1
test.vi	21.31	120	8

4.1.2. Monolingual Data

For monolingual data, we used English corpora of the WMT 2015,[6] which are permissible in the workshop IWSLT 2015. For Vietnamese data, we crawled articles from *wikipedia* by using more than 1.3B titles provided at *dumps.wikimedia.org*.[7] In addition, we crawled and extracted 800,000 Vietnamese articles from the website *baomoi.com*.[8] These articles were then pre-processed to produce a huge Vietnamese monolingual data. These monolingual data are shown in Table 3.

Table 3: Monolingual Data

Corpora	Sentences	Tokens
en	46,788,513	20,665,762
vi	21,180,758	1,960,909

4.1.3. Test Data

Test data of the workshop IWSLT 2015 include 1080 sentences on both English-Vietnamese and Vietnamese-English extracted from 12 talks of TED data. Statistics of sentences length of the test sets are shown in Table 2. The average length of the English and Vietnamese sets are 16.54 and 21.31, respectively. There are few sentences with length greater than 80.

[6]http://www.statmt.org/wmt15/translation-task.html
[7]http://dumps.wikimedia.org/viwiki/20150901/
[8]http://www.baomoi.com/

4.2. Experiments on Syntax-based Approaches

We found that advantages of syntax-based translation can resolve some differences between English and Vietnamese discussed in Section 2.1 including: i) reordering for syntactic reasons – e.g., move Vietnamese adjectives follow nouns ii) better explanation for function words – e.g., prepositions, determiners iii) conditioning to syntactically related words – translation of verbs may depend on subject or object.

Therefore, in our experiments, we attempted to apply syntax-based methods for the machine translation track. We used Joshua, a Java-based open source implementation of the hierarchical decoder (Li et al., 2009)[17], release 6.0.

Throughout this work, we applied two particular SCFG types known as Hiero (Chiang, 2007) and Syntax Augmented Machine Translation (SAMT) (Zollmann and Venugopal, 2006). We used Thrax (Weese, 2011) [14], an open-source grammar extractor for Hiero and SAMT grammars. We built systems for two language pairs for the IWSLT 2015 shared task: vi-en and en-vi. For the vi-en language pair, we built both SAMT and Hiero grammars, for the en-vi language pair, we only built Hiero grammar.

We used the constrained parallel data to train the translation models. The parallel data was subsampled using Joshua's built in subsampler to select sentences with n-grams relevant to the tuning and test sets. We used SRILM [29] to train a 5-gram LM with Kneser-Ney smoothing using the appropriate side of the parallel data. Before extracting an SCFG with Thrax, we pre-processed the data. For English side, we used the provided Perl scripts to tokenize and normalize the data. For Vietnamese side, we used JvnTextPro to tokenize data. We lower-case data before extracting an SCFG. For SAMT grammar extraction, we parsed the English training data using the Berkeley Parser (Petrov et al., 2006) [27] with the provided Treebank-trained grammar. We tuned the model weights against the tuning sets of the workshop using ZMERT (Zaidan, 2009) [33], an implementation of minimum error-rate training included with Joshua. We decoded the test set to produce a 300-best list of unique translations, then chose the best candidate for each sentence using Minimum Bayes Risk reranking (Kumar and Byrne, 2004) [28]. To re-case the 1-best test set output, we trained a true-case 5-gram LM using the same previous LM training data, and used Perl script to translate from the lowercased to true-case output. Table 4 shows experimental results of the submitted systems (phrase-based) and the syntax-based on the development set.

Table 4: Experimental results on the tuning data (BLEU)

Setup	en-vi	vi-en
SAMT	–	9.91
Hiero	19.27	12.52
Phrase-based (in-domain)	23.92	12.94
Phrase-based (out-of-domain)	25.49	18.27

We use BLEU [26] as the metric to evaluate our systems. Experimental results in Table 4 show higher BLEU scores of the phrase-based compared with the syntax-based methods on the development data. This kind of data, spoken languages, includes complicated structure sentences that we will discuss in Section 5. These characteristics lead to challenges for the syntax-based methods in parsing sentences into syntax structures. Since the results on development data, we set syntax-based outputs as contrastive runs, and phrase-based outputs are submitted to the workshop as primary runs.

4.3. Experiments on Domain Adaptation

Since we used unconstrained bilingual data from other domains in the phrase-based method, we attempted to apply some strategies for domain adaptation including fill-up and back-off combinations. We show experimental results of domain adaptation in this section.

Fill-up Combination (Bisazza et al., 2011 IWSLT): Fillup preserves all the entries and scores coming from the first model, and adds entries from the other models only if new. Moreover, a binary feature is added for each additional table to denote the provenance of an entry. These binary features work as scaling factors that can be tuned directly by MERT [24] along with other models' weights.

Back-Off Combination: This is a simplified version of fill-up. Nevertheless back-off technique does not generate the binary feature denoting the provenance an entry, and this makes the main advantage of back-off: the combined table contains the exact number of scores of their combining tables.

Table 5: Experimental results on domain-adaptation techniques (BLEU). Domain-adaptaion techniques: *fill-up* and *back-off*. *Merged-data*: merging in-domain and out-of-domain data for training.

Setup	en-vi	vi-en
Fill-up	27.90	17.68
Back-off	28.08	17.74
Merged-data	28.32	22.02

We compared results produced by fill-up and back-off techniques with those of the setup *merged-data* in which we merge all in-domain and out-of-domain data together and then train a translation model to obtain only one phrase table. Experimental results in Table 5 show higher scores in the merged-data training setup. Therefore, the merged-data setup was used for generating the primary runs.

4.4. Results

To train translation models, we merged the constrained and unconstrained bilingual corpora, then we run the processing steps described in Section 3. Table 6 shows BLEU scores of our translations on the evaluation system of the workshop.[9]

Table 6: Experimental results on the test sets IWSLT 2015 (BLEU). Hiero, SAMT: syntax-based systems. Submitted system: Phrase-based (out-of-domain).

Setup	en-vi	vi-en
baseline	27.01	24.61
SAMT	–	15.16
Hiero	21.48	15.05
Phrase-based (in-domain)	26.57	16.51
Phrase-based (out-of-domain)	28.17	21.53

We investigated and experimented syntax-based approaches using SAMT and Hiero grammars, which are described in Section 4.3. We used in-domain data for these systems. For English to Vietnamese translation (en-vi translation), Hiero shows a BLEU score of 21.48 which is 5.09 lower than the phrase-based method (26.57). For Vietnamese-English translation (vi-en translation), the result of Hiero is 1.46 lower than that of the phrase-based method (15.05 vs. 16.51). Meanwhile, SAMT which is experimented only on vi-en translation shows a slightly higher score than that of Hiero (15.16 vs. 15.05). For both translation directions, the phrase-based systems show higher BLEU scores than the syntax-based systems. The submitted system, phrase-based (out-of-domain), shows the highest BLEU scores (28.17 for en-vi translation, and 21.53 for vi-en translation). In comparison with the phrase-based in-domain system, the phrase-based out-of-domain system obtains higher BLEU scores (+1.6 for en-vi and +5.02 for vi-en translations) because of the supplemented data.

In comparison of translation directions, all systems show higher BLEU scores in en-vi than vi-en translations. In the result of Hiero, BLEU score of en-vi translation is 6.43 higher than that of vi-en translation. Similarly, the higher BLEU scores are +10.06 (phrase-based in-domain) and +6.64 (phrase-based out-of-domain).

In comparison with the baseline system of the workshop, our en-vi system shows the better result (28.17 vs. 27.01). Nevertheless, our vi-en system is worse than the baseline (21.53 vs. 24.61).

We will discuss these experimental results in the section of error analyses (Section 6).

5. Data Analysis

The data for machine translation track of the IWSLT 2015 are subtitles from TED talks. Since these data are in spoken language, there are some challenges for translation on this kind of data. We discuss several challenges with some examples.

[9] http://iwslt-server.fbk.eu/eval/Eval.html

<title> Rachel Pike: The science behind a climate headline </title>
Recently the headlines looked like this when the Intergovernmental Panel on **Climate Change**, or IPCC, put out their **report** on the state of **understanding of the atmospheric system**.
That report was written by 620 scientists from 40 countries.
They wrote almost a thousand pages on **the topic**.

Figure 1: An example of relationships in contexts and topics between sentences of TED data, emphasis (bold) added by author.

5.1. Context and Topic

The first problem is that there exists a connection between different sentences in a text. Sentences in a TED talk's subtitles may be related to each other in terms of context and topic. As shown in Figure 1, phrases *that report* and *the topic* are mentioned previously, and this can be seen as a dependent relationship between sentences. This kind of data causes the translation task more complicated than that of written texts in general.

5.2. Abstract Meaning

A characteristic of spoken languages like TED data is abstract meaning. As shown in Figure 2, *closet* does not mean *a cupboard or wardrobe*. Speakers sometimes tend to use metaphors in their speech, and it is not easy for machine translation systems to correctly produce output. This is also another challenge in translation tasks for TED data.

<title> Ash Beckham: We're all hiding something. Let's find the courage to open up </title>
<seg id="1"> I think we all have **closets**. </seg>
<seg id="2"> Your **closet** may be telling someone you love her for the first time, or telling someone that you're pregnant, or telling someone you have cancer, or any of the other hard conversations we have throughout our lives. </seg>

Figure 2: An example of abstract meaning in TED data, emphasis (bold) added by author.

5.3. Sentence Structures

Unlike written texts, structures of sentences in TED data are usually quite complicated, and this is a particular characteristic of spoken languages. This is not easy to realize and parse syntactic structures for sentences accurately. This also leads to the applying of syntax-based approaches for this kind of data more difficult. Figure 3 shows an example of this challenge.

<title> Mary Lou Jepsen: Could future devices read images from our brains? </title>
<seg id="14"> But that experience, I think, gave me a new appreciation for men and what they might walk through, and I've gotten along with men a lot better since then. </seg>

Figure 3: An example of complicated sentence structures in TED data.

Table 7: Out Of Vocabulary Statistics (%)

Setup	**en-vi**	**vi-en**
SAMT	–	1.67
Hiero	6.90	2.44
Phrase-based (in-domain)	5.03	2.50
Phrase-based (out-of-domain)	2.97	1.34

6. Errors Analysis

6.1. Out Of Vocabulary

We show statistics of out-of-vocabulary (OOV) of our systems on the test sets *tst2015*, which are described in Table 7. This is ratio of vocabulary of the test sets that cannot be translated by our systems to produce hypotheses. For the en-vi translations, that ratio of the Hiero (6.90%) is higher than that of the phrase-base in-domain (5.03%). The lowest OOV ratio is of the phrase-based (out-of-domain) which uses unconstrained data. This is also similar to the case of vi-en translations of the phrase-based (out-of-domain). Nevertheless, in the SAMT for vi-en translation, though the OOV ratio is lower than that of the phrase-based (in-domain) (1.67 % vs. 2.50 %), the SAMT still obtains a lower BLEU score (15.16 vs. 16.51). The systems may produce output phrases that differ from reference phrases even when input phrases are included in phrase tables. We discuss some examples of this problem in Section 6.2.

6.2. Hypotheses and Reordering

In Table 8 and Figure 4 two examples of translations are reported, analyzed in the following. In Table 8, we indicate some problems in vi-en translation in terms of meaning and tenses. The input phrase *được nhìn nhận* is translated into *was seen* (phrase-based), *visible* (syntax-based), *has been viewed* (reference, we use the file input of vi-en translation as the reference for vi-en translations). For another example, the input phrase *có thể suy nghĩ* is translated into *could think* (phrase-based and syntax-based), *can think* (reference). As we previously discussed in Section 2.1, Vietnamese differs from English in that it does not morphologically mark tenses. In this example, the two Vietnamese phrases are translated into results which are different from the reference. Recognizing tenses is a challenge for translation systems. This factor can be seen as a reason why

Table 8: An example of Vietnamese to English translation, bold phrases discussed in Section 6.2.

Input	<**title**>**Alex Wissner-Gross: A new equation for intelligence**</**title**> <**seg id="2"**>Nếu chúng ta nhìn lại lịch sử xem trí thông minh **được nhìn nhận** thế nào **ta có thể tham khảo** câu nói nổi tiếng của Edsger Dijsktra: "**Hỏi rằng** liệu máy **có thể suy nghĩ** được hay không cũng **thú vị** như hỏi **liệu một chiếc tàu ngầm có bơi được hay không**."</**seg**>
Phrase-based Output	If we look back in history to see the intelligence **was seen**, **we can refer to** the famous Edsger Dijsktra: **asking** whether machines **could think** or as **exciting** as the question of **whether a submarine had to swim or not**,
Hiero Output	If we look at what intelligence **visible** like, **we can go** even famous saying edsger_dijsktra history: "**or not asking** if machine **could think** is about as **exciting** as asked **if a submersible swimming or not**. "
Reference	If we take a look back at the history of how intelligence **has been viewed**, **one seminal example** has been Edsger Dijkstra's famous quote that "**the question of** whether a machine **can think** is about as **interesting** as the question of **whether a submarine can swim**."

vi-en translations show a lower performance than that of en-vi translation.

Another problem we would like to discuss in this example is choosing appropriate hypotheses. The input phrase *ta có thể tham khảo* is translated into *we can refer* (phrase-based), or *hỏi rằng* is translated into *asking* (phrase-based); *thú vị* is translated into *exciting* (phrase-based, syntax-based). These hypotheses can be accepted in terms of appropriate meaning. Nevertheless, they may be not matched with results of the reference: *one seminar example*, *the question of*, *interesting*, respectively. Therefore, choosing an appropriate hypothesis is another problem that should be solved to improve the translation performance.

In experimental results shown in Table 4 and Table 6, the syntax-based systems show lower scores than the phrase-based systems. Syntax-based methods may less appropriate for this kind of data, TED talks, than phrase-based methods. Nevertheless, we will show here an example that the syntax-based system produces a better result than the phrase-based in terms of reordering. In Figure 4, we describe translations of an English input sentence in the test set with the reference, phrase-based and syntax-based systems, respectively. The input noun phrase *This tool use ability* is translated by the phrase-based and syntax-based systems with different order in output phrases. The input phrase *use ability* which precedes the verb *will have* is a part of the subject, but its translation produced by the phrase-based system follows the verb and now becomes an object of the verb. This causes an incorrect meaning of the output. Meanwhile, this translation of the syntax-based system matches with the reference due to the syntactic analysis in syntax-based methods.

7. Conclusion

In this work, we have described the submitted system of the JAIST-UET-MITI team for the machine translation track of the IWSLT 2015 workshop. This year, we participated in the shared task for the language pair: English-Vietnamese. We investigated and experimented some approaches including phrase-based, syntax-based and domain adaptation. The submitted system, phrase-based approach, is based on the Moses toolkit, which shows the best results on both development sets and test sets in comparison with applied approaches. Although applying some domain adaptation techniques does not improve our unconstrained systems, we will attempt to deal with this by other strategies to obtain better results.

We have discussed some challenges of machine translation for the data domain of the shared task, subtitles of TED talks. We have also analyzed translation errors in some aspects in both approaches: phrase-based and syntax-based. We plan to deal with these issues in the future work.

8. Acknowledgements

This paper has been supported by the project "Xây dựng hệ thống dịch tự động hỗ trợ việc dịch các tài liệu giữa tiếng Việt và tiếng Nhật nhằm giúp các nhà quản lý và doanh nghiệp Hà Nội tiếp cận và làm việc hiệu quả với thị trường Nhật Bản" funded by Hanoi Department of Science and Technology.

9. References

[1] Ho Tu Bao, Pham Ngoc Khanh, Ha Thanh Le, and Nguyen Thi Phuong Thao. Issues and first development phase of the english-vietnamese translation system evsmt1. 0. 2008.

[2] Arianna Bisazza, Nick Ruiz, Marcello Federico, and FBK-Fondazione Bruno Kessler. Fill-up versus interpolation methods for phrase-based smt adaptation. In *IWSLT*, pages 136–143, 2011.

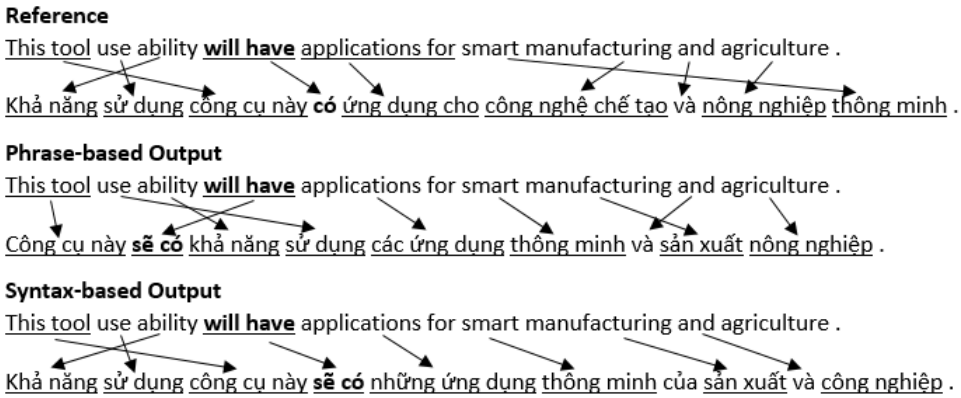


Figure 4: An example of reordering discussed in Section6.2: translations of the reference, phrase-based and syntax-based, respectively. The bold phrases indicate the verbs of the sentences.

[3] Mauro Cettolo, Christian Girardi, and Marcello Federico. Wit3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, 2012.

[4] Colin Cherry and George 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*, pages 427–436. Association for Computational Linguistics, 2012.

[5] David Chiang. Hierarchical phrase-based translation. *computational linguistics*, 33(2):201–228, 2007.

[6] Dien Dinh. Building a training corpus for word sense disambiguation in english-to-vietnamese machine translation. In *Proceedings of the 2002 COLING workshop on Machine translation in Asia-Volume 16*, pages 1–7. Association for Computational Linguistics, 2002.

[7] Dien Dinh and Hoang Kiem. Pos-tagger for english-vietnamese bilingual corpus. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond-Volume 3*, pages 88–95. Association for Computational Linguistics, 2003.

[8] Dien Dinh, Hoang Kiem, and Eduard Hovy. Btl: a hybrid model for english-vietnamese machine translation. In *Proceedings of the MT Summit IX*, pages 23–27, 2003.

[9] Dien Dinh, Nguyen Luu Thuy Ngan, and Van Chi Nam Do Xuan Quang. A hybrid approach to word order transfer in the english-to-vietnamese machine translation. In *Proceedings of the MT Summit IX*, 2003.

[10] Michel Galley and Christopher D Manning. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 848–856. Association for Computational Linguistics, 2008.

[11] Qin Gao and Stephan Vogel. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57. Association for Computational Linguistics, 2008.

[12] Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics, 2011.

[13] Vu Hoang, Mai Ngo, and Dien Dinh. A dependency-based word reordering approach for statistical machine translation. In *Research, Innovation and Vision for the Future, 2008. RIVF 2008. IEEE International Conference on*, pages 120–127. IEEE, 2008.

[14] Weese Jonathan. A systematic comparison of synchronous contextfree grammars for machine translation. Master's thesis, Johns Hopkins University, 2011.

[15] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[16] Philipp Koehn, Franz Josef Och, and Daniel 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, pages 48–54. Association for Computational Linguistics, 2003.

[17] Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren NG Thornton, Jonathan Weese, and Omar F Zaidan. Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139. Association for Computational Linguistics, 2009.

[18] Quoc Hung Ngo, Dinh Dien, and Werner Winiwarter. Automatic searching for english-vietnamese documents on the internet. In *24th International Conference on Computational Linguistics*, page 211, 2012.

[19] Quoc Hung Ngo and Werner Winiwarter. Building an english-vietnamese bilingual corpus for machine translation. In *Asian Language Processing (IALP), 2012 International Conference on*, pages 157–160. IEEE, 2012.

[20] Quoc Hung Ngo, Werner Winiwarter, and Bartholomäus Wloka. Evbcorpus-a multi-layer english-vietnamese bilingual corpus for studying tasks in comparative linguistics. In *Proceedings of the 11th Workshop on Asian Language Resources (11th ALR within the IJCNLP2013)*, pages 1–9, 2013.

[21] Thai Phuong Nguyen and Akira Shimazu. Improving phrase-based statistical machine translation with morphosyntactic transformation. *Machine Translation*, 20(3):147–166, 2006.

[22] Thai Phuong Nguyen and Akira Shimazu. A syntactic transformation model for statistical machine translation. In *Computer Processing of Oriental Languages. Beyond the Orient: The Research Challenges Ahead*, pages 63–74. Springer, 2006.

[23] Thai Phuong Nguyen, Akira Shimazu, Tu-Bao Ho, Minh Le Nguyen, and Vinh Van Nguyen. A tree-to-string phrase-based model for statistical machine translation. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 143–150. Association for Computational Linguistics, 2008.

[24] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.

[25] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003.

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[27] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440. Association for Computational Linguistics, 2006.

[28] Kumar Shankar and Byrne William. 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.

[29] Andreas Stolcke et al. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*, 2002.

[30] Giang Tang. Cross-linguistic analysis of vietnamese and english with implications for vietnamese language acquisition and maintenance in the united states. *Journal of Southeast Asian American Education and Advancement*, 2(1):3, 2007.

[31] Nguyen Chanh Thanh, Nguyen Chi Hieu, Phan Thi Tuoi, et al. Dictionaries for english-vietnamese machine translation. In *Computer Processing of Oriental Languages. Beyond the Orient: The Research Challenges Ahead*, pages 363–369. Springer, 2006.

[32] Vinh Van Nguyen, Thai Phuong Nguyen, Akira Shimazu, and Minh Le Nguyen. Reordering phrase-based machine translation over chunks. In *Research, Innovation and Vision for the Future, 2008. RIVF 2008. IEEE International Conference on*, pages 114–119. IEEE, 2008.

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

[34] Andreas Zollmann and Ashish Venugopal. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141. Association for Computational Linguistics, 2006.

PJAIT Systems for the IWSLT 2015 Evaluation Campaign Enhanced by Comparable Corpora

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

In this paper, we attempt to improve Statistical Machine Translation (SMT) systems on a very diverse set of language pairs (in both directions): Czech - English, Vietnamese - English, French - English and German - English. To accomplish this, we performed translation model training, created adaptations of training settings for each language pair, and obtained comparable corpora for our SMT systems. Innovative tools and data adaptation techniques were employed. The TED parallel text corpora for the IWSLT 2015 evaluation campaign were used to train language models, and to develop, tune, and test the system. In addition, we prepared Wikipedia-based comparable corpora for use with our SMT system. This data was specified as permissible for the IWSLT 2015 evaluation. We explored the use of domain adaptation techniques, symmetrized word alignment models, the unsupervised transliteration models and the KenLM language modeling tool. To evaluate the effects of different preparations on translation results, we conducted experiments and used the BLEU, NIST and TER metrics. Our results indicate that our approach produced a positive impact on SMT quality.

1. Introduction

Statistical Machine Translation (SMT) must deal with a number of problems to achieve high quality. These problems include the need to align parallel texts in language pairs and cleaning harvested parallel corpora to remove errors. This is especially true for real-world corpora developed from text harvested from the vast data available on the Internet. Out-Of-Vocabulary (OOV) words must also be handled, as they are inevitable in real-world texts [1].

The lack of enough parallel corpora is another significant challenge for SMT. Since the approach is statistical in nature, a significant amount of quality language pair data is needed to improve translation accuracy. In addition, very general translation systems that work in a general text domain have accuracy problems in specific domains. SMT systems are more accurate on corpora from a domain that is not too wide. This exacerbates the data problem, calling for the enhancement of comparable corpora for particular text domains [2].

This paper describes SMT research that addresses these problems, particularly comparable corpora within the limits of permissible data for the IWSLT 2015 campaign. We selected a diverse set of language pairs for translation in both directions: Czech and English, Vietnamese and English, French and English, and German and English. To accomplish this, we performed model training, created adaptations of training settings and data for each language pair, and enhanced our systems by building and using comparable corpora in our statistical translation systems.

Innovative tools and data adaptation techniques were employed. The Technology, Entertainment and Design (TED) parallel text corpora for the IWSLT 2015 evaluation campaign were used to train language models, and to develop, tune, and test the system. In addition, we prepared Wikipedia-based comparable corpora for use with our SMT systems. We explored the use of domain adaptation techniques, symmetrized word alignment models, the unsupervised transliteration models, and the KenLM language modeling tool [3]. To evaluate the effects of different preparations on translation results, we conducted experiments and evaluated the results using standard SMT metrics [4].

The languages translated during this research are diverse: Czech, English, French, German, and Vietnamese. The first four belong to three different branches of the Indo-European language family. Czech is found in the Slavic branch of that language family. English and German fall in the Western group of the Germanic branch of the Indo-European family, while French is found in the Romance branch. Vietnamese falls into an entirely different language family, Austro-Asiatic. So, our translation challenges cross languages, language branches, and language families [5, 6, 7, 8, 9].

This paper is structured as follows: Section 2 explains the data preparation. Section 3 presents experiment setup and the results. Lastly in Section 4 we summarize the work.

2. Data preparation

This section describes our techniques for data preparation for our SMT systems. We give particular emphasis to preparation of the language data and models and our domain adaptation approach.

2.1. Obtaining Comparable Corpora

We provided a new approach for mining parallel corpora from Wikipedia in support of IWSLT 2015 objectives. In general, Wikipedia data is noisy, has a wide domain, and the sentences of its bilingual texts are not aligned. To extract additional data from Wikipedia, we used the method described in [17] and adapted it for the needs of new languages.

Our tool performs model training and phrase-level symmetrization in a multi-threaded approach to increase performance. The dedicated tuning tool is used to determine the trained model's optimal weights [17].

To be more precise our approach enhances the Yalign tool [18], which is single threaded and becomes computationally infeasible for large-scale mining of corpora. Yalign's existing classifier also requires individual pairs of input text or webpages to be loaded into memory for alignment, and the classifier must be reloaded. To increase performance, we

modified the design to supply the classifier with Wikipedia articles within one session, with no need to reload the classifier. Performance improvements of more than a factor of 6x have been observed, as a result [17].

Our approach also replaces Yalign's alignment algorithm with a GPU-optimized version of the Needleman-Wunsch algorithm. In addition to the performance improvements this brings, it also yields higher alignment accuracy. A tuning algorithm enables automatic selection of threshold and penalty parameters to adaptably perform tradeoffs between precision and recall [17].

Wikipedia data in the following language pairs were mined using our tool: English-Czech, English-German, English-French, and English-Vietnamese. This dataset was accepted by the IWSLT 2015 evaluation organizers as permissible data [19].

2.2. Data Preparation

Five languages were involved in this research: Czech, English, French, German, and Vietnamese. TED talks training data for those languages consisted of the following:

- Czech: approx. 11 MB - 176,094 untokenized words
- French: approx. 24 MB - 165,605 untokenized words
- German: approx. 22 MB - 213,486 untokenized words
- Vietnamese: approx. 18 MB, 52,549 untokenized words
- English: approx. 125,000 untokenized words for DE and FR, 85,000 words for CS and 100,000 for VI pairs

The TED transcripts prepared by the FBK team for IWSLT[1] consist of text encoded in UTF-8 format, separated into sentences, and provided in pairs of languages. The data is provided as XML files [1].

Pre-processing, both automatic and manual, of this training data was required. There were a variety of errors found in this data, including spelling errors, unusual nesting of text, text duplication, and parallel text issues. Approximately 2% of the text in the training set contained spelling errors, and approximately 4% of the text had insertion errors. A tool described in [2], was used to correct these errors. Previous studies have found that such cleaning increases the BLEU score for SMT by a factor of 1.5–2 [1].

After cleaning and tokenization, we found the following amounts of unique word forms in the different languages:

- Czech: 109,692 tokenized words
- French: 76,216 tokenized words
- German: 123,673 tokenized words
- Vietnamese: 24,914 tokenized words
- English: approx. 53,000 tokenized words for DE and FR, 39,000 tokenized words for CS and 44,000 for VI pairs

In addition, comparable corpora in those language pairs were created from Wikipedia data. The Wikipedia data obtained consisted of the following:

- Czech: approx. 22 MB - 104,698 untokenized words in 27,723 parallel sentences
- French: approx. 101 MB – 1,121,424 untokenized words in 818,300 parallel sentences
- German: approx. 277 MB – 2,865,865 untokenized words in 2,459,662 parallel sentences
- Vietnamese: approx. 33 MB - 93,218 untokenized words in 58,116 parallel sentences
- English: approx. 285MB – 2,577,854 untokenized words in DE pair, approx. 113MB – 1,290,499 untokenized words in FR pair, approx. 22MB – 98,820 untokenized words in CS pair and approx. 34MB – 92,445 untokenized words in VI pair

After cleaning [2] and tokenization:

- Czech: approx. 73,996 words in CS side and 65,592 words at EN side
- French: approx. 517,604 words in FR side and 544,994 words at EN side
- German: approx. 1,327,789 words in DE side and 922,785 words at EN side
- Vietnamese: approx. 66,391 words in FR side and 65,852 words at EN side

SyMGiza++, a tool that supports the creation of symmetric word alignment models, was used to extract parallel phrases from the data. This tool enables alignment models that support many-to-one and one-to-many alignments in both directions between two language pairs. SyMGiza++ is also designed to leverage the power of multiple processors through advanced threading management, making it very fast. Its alignment process uses four different models during training to progressively refine alignment results. This approach has yielded impressive results in [10].

Out-Of-Vocabulary (OOV) words pose another significant challenge to SMT systems. If not addressed, unknown words appear, untranslated, in the output, lowering the translation quality. To address OOV words, we used implemented in the Moses toolkit Unsupervised Transliteration Model (UTM). UTM is an unsupervised, language-independent approach for learning OOV words. We used the post-decoding transliteration option with this tool. UTM uses a transliteration phrase translation table to evaluate and score multiple possible transliterations [11, 12].

The KenLM tool was applied to the language model to train and binarize it. This library enables highly efficient queries to language models, saving both memory and computation time. The lexical values of phrases are used to condition the reordering probabilities of phrases. We used KenLM with lexical reordering set to hier-msd-bidirectional-fe. This setting uses a hierarchical model that considers three orientation types based on both source and target phrases: monotone (M), swap (S), and discontinuous (D). Probabilities of possible phrase orders are examined by the bidirectional reordering model [3, 13, 14].

2.3. Domain Adaptation

The TED data sets have a rather a wide domain, but rather not as wide-ranging in topic as the Wikipedia articles. Since SMT systems work best in a defined domain, this presents another considerable challenge. If not addressed, this would lead to lower translation accuracy.

The quality of domain adaptation depends heavily on training data used to optimize the language and translation models in an SMT system. Selection and extraction of domain-specific training data from a large, general corpus addresses this issue [15]. This process uses a parallel, general domain corpus and a general domain monolingual corpus in

[1] iwslt.org

the target language. The result is a pseudo in-domain sub-corpus.

As described by Wang et al. in [16], there are generally three processing stages in data selection for domain adaptation. First, sentence pairs from the parallel, general domain corpus are scored for relevance to the target domain. Second, resampling is performed to select the best-scoring sentence pairs to retain in the pseudo in-domain sub-corpus. Those two steps can also be applied to the general domain monolingual corpus to select sentences for use in a language model. After collecting a substantial amount of sentence pairs (for the translation model) or sentences (for the language model), those models are trained on the sub-corpus that represents the target domain [16].

Similarity measurement is required to select sentences for the pseudo in-domain sub-corpus. There are three state-of-the-art approaches for similarity measurement. The cosine tf-idf criterion looks for word overlap in determining similarity. This technique is specifically helpful in reducing the number of OOV words, but it is sensitive to noise in the data. A perplexity-based criterion considers the *n*-gram word order in addition to collocation. Lastly, edit distance simultaneously considers word order, position, and overlap. It is the strictest of the three approaches. In their study [16], Wang et al. found that a combination of these approaches provided the best performance in domain adaptation for Chinese-English corpora [16].

In accordance with Wang et al.'s approach [16], we use a combination of the criteria at both the corpora and language models. The three similarity metrics are used to select different pseudo in-domain sub-corpora. The sub-corpora are then joined during resampling based on a combination of the three metrics. Similarly, the three metrics are combined for domain adaptation during translation. We empirically found acceptance rates that allowed us only to harvest 20% of most domain-similar data [16].

3. Experimental Results

Various versions of our SMT systems were evaluated via experimentation. In preparation for experiments, we processed the corpora. This involved tokenization, cleaning, factorization, conversion to lower case, splitting, and final cleaning after splitting. Language models were developed and tuned using the training data.

The Experiment Management System [4] from the open source Moses SMT toolkit was used to conduct the experiments. Training of a 6-gram language model was accomplished our resulting systems using the KenLM Modeling Toolkit instead of 5-gram SRILM [20] with an interpolated version of Kneser-Key discounting (interpolate –unk –kndiscount) that was used in our baseline systems. Word and phrase alignment was performed using SyMGIZA++ [10] instead of GIZA++. KenLM was also used, as described earlier, to binarize the language models. The OOV's were handled by using Unsupervised Transliteration Model [12].

The results are shown in Table 1 and 2. Each language pair was translated in both directions. "BASE" in the tables represents the baseline SMT system. "EXT" indicates results for the baseline system, using the baseline settings but extended with comparable corpora from Wikipedia. "BEST" indicates the results when the new SMT settings were applied and using all permissible data. For DE and FR we did not train systems using more permissible data that our Wikipedia comparable corpora. Additionally, we conducted progressive tests only for FR and DE data because CS and VI were not evaluated before during IWSLT campaigns.

Three well-known metrics were used for scoring the results: Bilingual Evaluation Understudy (BLEU), the US National Institute of Standards and Technology (NIST) metric and Translation Error Rate (TER).

In addition to TED data, the data permissible for the IWSLT 2015 campaign included: data from the Workshop on Machine Translation (WMT) 2015 web page [21], MultiUN data [22, 23] (translated United Nations documents) and parallel corpora we provided from the Wikipedia [19].

The results show that the systems extended with comparable corpora from Wikipedia performed well on all data sets in comparison to the baseline SMT systems. Application of the new settings and use of all permissible data improved performance even more.

Table 1: Progressive Results, 2014 Test Data

LANG	SYSTEM	DIRECTION	BLEU	NIST	TER
DE-EN	BASE	→EN	17.99	5.51	64.35
	EXT	→EN	21.92	6.04	60.58
	BASE	←EN	18.49	5.74	61.65
	EXT	←EN	20.68	5.99	59.77
FR-EN	BASE	→EN	32.20	7.36	47.60
	EXT	→EN	32.92	7.37	48.25
	BASE	←EN	30.31	7.24	50.17
	EXT	←EN	31.88	7.49	47.92

Table 2: Results, 2015 Test Data

LANG	SYSTEM	DIRECTION	BLEU	NIST	TER
DE-EN	BASE	→EN	21.78	6.49	55.45
	EXT	→EN	26.08	7.03	54.34
	BASE	←EN	20.08	5.76	61.37
	EXT	←EN	22.51	6.04	59.02
FR-EN	BASE	→EN	31.94	7.34	47.55
	EXT	→EN	32.75	7.27	48.40
	BASE	←EN	30.54	6.99	51.51
	EXT	←EN	32.79	7.32	49.15
CS-EN	BASE	→EN	22.44	6.11	57.98
	EXT	→EN	24.19	6.03	56.13
	BEST	→EN	25.07	6.40	55.74
	BASE	←EN	14.74	4.74	65.80
	EXT	←EN	15.18	4.86	65.11
	BEST	←EN	17.17	5.10	63.00
VI-EN	BASE	→EN	24.61	5.92	59.32
	EXT	→EN	22.41	5.68	63.78
	BEST	→EN	23.46	5.73	62.19
	BASE	←EN	27.01	6.47	58.42
	EXT	←EN	27.16	6.23	66.18
	BEST	←EN	28.39	6.67	65.01

4. Summary

We have improved SMT for a very diverse set of language pairs, in both translation directions, using data permissible for the IWSLT 2015 evaluation campaign. We cleaned, prepared, and tokenized the training data. Symmetric word alignment models were used to align the corpora. UTM was used to handle OOV words. A language model was created, binarized, and tuned. We performed domain adaptation of language data using a combination of similarity metrics.

Experiments were performed using the data permissible by the IWSLT 2015 organizers. The results show a positive impact of our approach on SMT quality across the language

pairs. Only surprising result was in translation from Vietnamese into English, where our best system outperformed the baseline. We conducted detailed research regarding this issue, including tuning results for each iteration and evaluation of each TED talk separately. We found out that on most talks our system worked correctly only two of them the results were negative. The talk number 2183 the baseline BLEU score was equal to 63.88 (BASE) whereas our system score (BEST) was equal to 49.73. Such big disproportion is most likely reason for strange overall evaluation score. We believe that some parts of talk 2183 were present in training data, and extending this data decreased the scores. Detailed evaluation results are presented in the Table 3. Additionally, we can suspect, from the statistics presented in Chapter 2.2, that Wikipedia data for Vietnamese is was not good enough. Having 93,218 words in 58,116 sentences would mean that this corpus basically consists of uni- or bi-grams.

Table 3: Detailed Vietnamese-English Results

TALK ID	SYSTEM	BLEU
1922	BASE	17.25
	BEST	18.07
1932	BASE	15.70
	BEST	18.17
1939	BASE	12.35
	BEST	13.26
1954	BASE	27.96
	BEST	28.92
1961	BASE	19.59
	BEST	21.35
1997	BASE	16.81
	BEST	19.67
2007	BASE	16.38
	BEST	19.67
2017	BASE	21.08
	BEST	22.42
2024	BASE	9.44
	BEST	6.25
2045	BASE	21.05
	BEST	21.09
2102	BASE	17.14
	BEST	20.16
2183	BASE	63.88
	BEST	49.73

5. References

[1] Wołk, K.; Marasek, K. Polish - English Speech Statistical Machine Translation Systems for the IWSLT 2014, In: *Proceedings of International Workshop on Spoken Language Translation*, Lake Tahoe, California, USA, 2014, pp. 143-148.

[2] Wołk, K.; Marasek, K. A Sentence Meaning Based Alignment Method for Parallel Text Corpora Preparation. In: *New Perspectives in Information Systems and Technologies, Volume 1*. Springer International Publishing, 2014. p. 229-237.

[3] Heafield, K. KenLM: Faster and smaller language model queries, In: *Proc. of Sixth Workshop on Statistical Machine Translation*, Association for Computational Linguistics, 2011.

[4] P. Koehn et al., Moses: Open Source Toolkit for Statistical Machine Translation, In: *Proceedings of the ACL 2007 Demo and Poster Sessions*, Prague, June 2007, pp. 177–180.

[5] The Technology Development Group, "Czech." Last revised March 21, 2014. Retrieved September 27, 2015 from: aboutworldlanguages.com/czech

[6] The Technology Development Group, "English." Last revised October 14, 2013. Retrieved September 27, 2015 from: aboutworldlanguages.com/english

[7] The Technology Development Group, "French." Last revised January 21, 2014. Retrieved September 27, 2015 from: aboutworldlanguages.com/french

[8] The Technology Development Group, "German." Last revised November 7, 2014. Retrieved September 27, 2015 from: aboutworldlanguages.com/german

[9] The Technology Development Group, "Vietnamese." Last revised March 4, 2013. Retrieved September 27, 2015 from: aboutworldlanguages.com/vietnamese

[10] Junczys-Dowmunt, M.; Szał A.; SyMGiza++: symmetrized word alignment models for statistical machine translation. In: *Security and Intelligent Information Systems*. Springer Berlin Heidelberg, 2012. 379-390.

[11] Moses statistical machine translation, "OOVs." Last revised February 13, 2015. Retrieved September 27, 2015 from: http://www.statmt.org/moses/?n=Advanced.OOVs#ntoc2

[12] Durrani, N., et al.; Integrating an unsupervised transliteration model into statistical machine translation. In: *EACL 2014* (2014): 148.

[13] Costa-Jussa, M.; Fonollosa, J.; Using linear interpolation and weighted reordering hypotheses in the Moses system, Barcelona, Spain, 2010.

[14] Moses statistical machine translation, "Build reordering model." Last revised July 28, 2013. Retrieved October 10, 2015 from: http://www.statmt.org/moses/?n=FactoredTraining.BuildReorderingModel

[15] Axelrod, A.; He X.; Gao J., Domain adaptation via pseudo in-domain data selection. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*, pp. 355–362, July 2011.

[16] Longyue Wang, Derek F. Wong, Lidia S. Chao, Yi Lu, and Junwen Xing, "A Systematic Comparison of Data Selection Criteria for SMT Domain Adaptation., *The Scientific World Journal*, vol. 2014, Article ID 745485, 10 pages, 2014. doi:10.1155/2014/745485

[17] Wołk, K.; Marasek, K., Tuned and GPU-accelerated Parallel Data Mining from Comparable Corpora, In: *Lecture Notes in Artificial Intelligence*, p. 32 – 40, ISBN: 978-3-319-24032-9, Springer, 2015.

[18] Berrotarán G., Carrascosa R., Vine A., Yalign documentation, accessed 01/2015.

[19] "IWSLT Evaluation 2015." Last revised (n.d.), Retrieved September 27, 2015 from: iwslt.org

[20] Stolcke A., SRILM - An Extensible Language Modeling Toolkit., *INTERSPEECH*, 2002.

[21] EMNLP 2015 Tenth Workshop on Statistical Machine Translation, "Shared Task: Machine Translation." Last revised (n.d.), Retrieved September 27, 2015 from: http://www.statmt.org/wmt15/translation-task.html

[22] "MultiUN." Last revised (n.d.), Retrieved September 27, 2015 from: http://opus.lingfil.uu.se/MultiUN.php

[23] Tiedemann, J., Parallel Data, Tools and Interfaces in OPUS. In: *Proceedings of the 8th International Conference on Language Resources and Evaluation* (LREC 2012), 2012.

The NAIST English Speech Recognition System for IWSLT 2015

Michael Heck, Quoc Truong Do, Sakriani Sakti, Graham Neubig, Satoshi Nakamura

Augmented Human Communication Laboratory,
Graduate School of Information Science,
Nara Institute of Science and Technology,
Nara, Japan

{michael-h,do.truong.dj3,ssakti,neubig,s-nakamura}@is.naist.jp

Abstract

The International Workshop for Spoken Language Translation (IWSLT) is an annual evaluation campaign for core speech processing technologies. This paper presents Nara Institute of Science and Technology's (NAIST's) contribution to the English automatic speech recognition (ASR) track for the 2015 evaluation campaign. The ASR systems presented in this paper make use of various front-ends, varying deep neural net (DNN) acoustic models and separate language models for decoding and rescoring. Recognition is performed in three stages: Decoding, lattice rescoring and system combination via recognizer output voting error reduction (ROVER). We discuss the application of a rank-score based weighting approach for the system combination. Also, a Gaussian mixture model hidden Markov model (GMM-HMM) based speech/non-speech segmenter makes use of said combination scheme. The primary submission achieves a word error rate (WER) of 9.5% and 10.1% on the official development set, given manual and automatic segmentation respectively.

1. Introduction

The 2015 evaluation campaign of the 12th International Workshop on Spoken Language Translation (IWSLT) offers participants the opportunity to advance the state-of-the-art in core tasks of spoken language translation. This involves the tasks of automatic speech recognition (ASR), machine translation (MT) and the combination of ASR and MT, the task of spoken language translation (SLT) itself. All tasks are performed and evaluated on multi-topic TED (short for Technology, Entertainment, Design) and TEDx (licensed spin-off) conference talks (`http://www.ted.com`). This paper describes Nara Institute of Science and Technology's contribution to this year's evaluation campaign by participation in the ASR track for the English language. The goal of this track is the automatic transcription of unsegmented talks, thus the task is two-fold: automatic segmentation followed by speech recognition. We describe the development and application of a Gaussian mixture model (GMM) based speech/non-speech segmenter using the Janus speech recognition toolkit [1] (see Section 4) and the ASR system development and decoding utilizing the Kaldi speech recognition toolkit [2] (see Sections 2 and 5 respectively). Our speech-to-text system makes use of various front-ends, deep neural net (DNN) acoustic models and several language models for decoding and rescoring.

High performance speech recognition often makes use of system combination approaches, especially if recognition in real-time is not a major concern. Recognizer output voting error reduction (ROVER) [3] and confusion network combination (CNC) [4] are among the most popular methods. With confidence scores in hand, both techniques allow for some form of weighting, and studies [5, 6] have affirmed the advantages of confidence based weighting strategies. However, it is common practice that systems that contribute to a combination do so with equal shares: Besides the commonly applied word or segment based weighting, e.g. during lattice combination, systems usually contribute equally to the final output. This strategy however can fail in cases where system performances are unbalanced and better hypotheses might simply be overpowered by suboptimal alternatives. In previous work [7] we were able to show the positive effects of a weighted system combination method that makes use of weights on the system level. In this work, we expand this weighting technique to automatic segmentation by combining multiple models for the segmentation task, in addition to using the system combination for decoding.

2. Overall system

In this section, we describe the components of our framework and the details of the system development. We elaborate our usage of several acoustic front-ends, acoustic modeling, and language modeling. The general framework is illustrated in Fig. 1. The automatic segmentation is explained in the following section.

2.1. Acoustic features

We utilized three different kinds of acoustic features: *a)* Mel-frequency cepstral coefficients (MFCC) [8]; *b)* perceptual linear prediction (PLP) [9]; *c)* log Mel-filter bank (FBANK). All feature vector types are 40-dimensional (raw output without dimension reduction), and are extracted for every 10 ms with a window length of 25 ms.

Additionally, in order to enhance the input features, we also adopt i-vector features [10], which were originally proposed for the speaker identification task. The distribution of an utterance supervector M can be modeled by

$$M = m + Tw \tag{1}$$

where m is the mean distribution vector, T is a total variability matrix, and w is the i-vector. By having m and T fixed for all utterances, w would be affected by speaker and channel characteristics. We utilized i-vectors because they are able to capture speaker and channel informations that might be helpful for speech recognition, but are not represented in standard features such as MFCC, PLP, and FBANK.

2.2. Acoustic model training

We tested several acoustic model training strategies during development. GMM- and DNN-based acoustic models were trained with

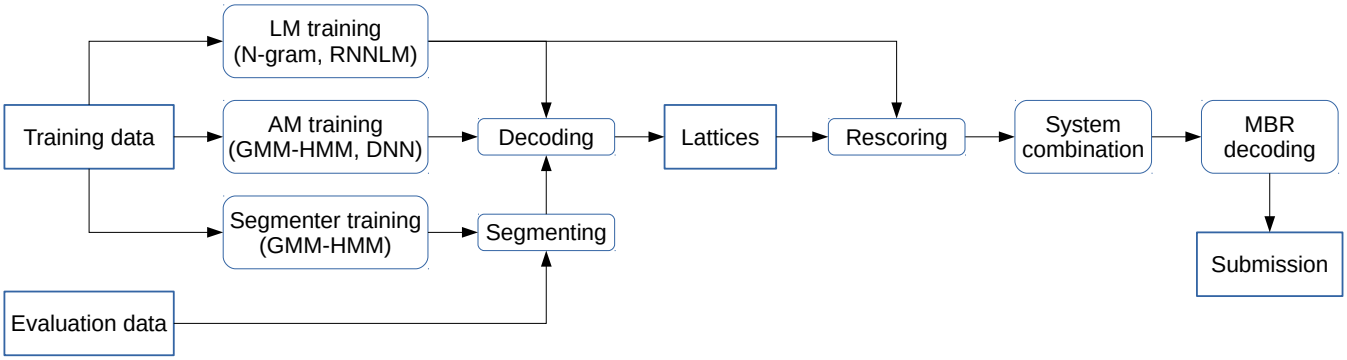


Figure 1: *General overview of our framework.*

different types of input features, as shown in Table 1. Models using speaker adaptive training (SAT) use standard features + feature-space maximum likelihood linear regression (fMLLR) [11], while all but one of the DNN-based acoustic models are trained with stacked standard and i-vector features. We investigated DNN architectures using *sigmoid*, *rectified linear* (ReLU), or *p-norm* [12] units, and also perform training using *state-level minimum Bayes risk* (sMBR) [13, 14]. The models are all implemented using the Kaldi speech recognition toolkit [2], and details are described in the following subsections.

2.2.1. Architectures

The *sigmoid DNN* model can be considered a standard DNN acoustic model with 6 hidden layers, where each layer consists of 2048 nodes. The sigmoid activation function is applied in each hidden layer, and the softmax function is applied in the output layer. The input features are generated by linear discriminant analysis (LDA) + maximum likelihood linear transform (MLLT) + fMLLR performed on top of spliced MFCC or FBANK (with splicing context 4). These feature vectors are also spliced with 5 preceding and 5 succeeding vectors, resulting in the final 440 dimensional DNN input feature vector covering 11 frames of context. First, we performed the pre-training with a restricted Boltzmann machine (RBM) deep belief network [15]. After that, the DNN was trained using the back-propagation algorithm and stochastic gradient descent with frame cross-entropy (CE) criterion as implemented by the Kaldi speech recognition toolkit [2].

We trained a *ReLU DNN* because it has been reported in [16] that rectified linear units can show better performance than sigmoid units for large vocabulary continuous speech recognition (LVCSR) tasks. We utilized a ReLU DNN with 6 hidden layers, where each layer consists of 1024 nodes, and the ReLU activation function is applied in each hidden layer. The input features are a raw 40 dimensional standard feature vector and a 100 dimensional i-vector stacked on top. Further, we do not perform pre-training as for the sigmoid DNN model, but instead we train for a fixed number of epochs and average model parameters over the last few epochs of training [17]. The parameters are also optimized according to the frame CE criterion.

The *p-norm DNN* [12] was adopted as the third type of model. The p-norm is a "dimension-reducing" non-linearity that is inspired by maxout

$$y = ||\mathbf{x}||_p = \left(\sum_i |x_i|^p \right)^{1/p}, \qquad (2)$$

where here the vector $\mathbf{x}$ represents a bundled set of 10 feature vectors, p is the normalized parameter and is set to 2 as it showed the best performance as described in [12]. The model architecture is the same with ReLU DNN with 6 hidden layers, each has 1024 nodes. The input features are also the same as for the ReLU DNN. The parameters are trained by using frame CE.

Note that for *ReLU DNN* and *p-norm DNN*, we perform feature splicing at the first, second, forth, and fifth layers with the following frame indexes,

- first layer: -2, -1, 0, 1, 2,
- second layer: -1, 2,
- forth layer: -3, 3,
- fifth layer: -7, 2.

2.2.2. sMBR training

To further enhance the DNN model, we continued training the model according to the state-level minimum Bayes risk (sMBR) criterion. This DNN is a p-norm DNN model but it is optimized according to sMBR instead of cross entropy. We only attempted to optimize the p-norm DNN this way because the training with sMBR is quite complicated and time-consuming, and more importantly, the p-norm DNN outperformed other models on "tst2013" test set during our experiments.

The training procedure is as follows: We first perform forced alignment, followed by a decoding on the training data to derive training samples, this process took 2 days on a cluster machine with 80 CPUs to produce 80 lattices. Then, we merge all lattices down to 4, which is equal to the number of GPUs we utilize. Finally, we perform parallel sMBR training as implemented in Kaldi.

2.3. Dictionary

We utilized a modified CMU pronouncing dictionary (`http://www.speech.cs.cmu.edu/cgi-bin/cmudict`) consisting of about 100k words as a base dictionary. We also employed grapheme-to-phoneme (G2P) conversion using the Sequitur G2P toolkit [18] trained on the CMU dictionary to generate pronunciations for unknown words in the training data. As a result, the total number of words in our dictionary is about 210k words. This dictionary is used for training as well as decoding.

2.4. Language model training

2.4.1. N-gram

N-grams have long been a standard language modeling technique for ASR, where $N-1$ words are used as context to predict the

Front-end	Model type				
	GMM-HMM (SAT)	Sigmoid DNN (CE)	ReLU DNN (CE)	p-norm DNN (CE)	p-norm DNN (sMBR)
MFCC	✓	✓	✓	✓	✓
PLP	✓	-	✓	✓	-
FBANK	✓	✓	✓	✓	✓

Table 1: *The list of all trained acoustic models.*

next word. The larger the context, the more data is required to avoid the data sparsity problem. For the experiments described here, two N-gram language models (LMs) were trained with Kneser-Ney smoothing [19] implemented in the SRILM toolkit [20], a 4-gram LM pruned with probability 10^{-8} for decoding purposes, and a full 5-gram model for rescoring in a second pass.

2.4.2. RNNLMs

Recurrent neural network language models (RNNLMs) have shown to have an advantage over the standard N-gram language model. There are several reasons for this, perhaps the most notable being that RNNLMs can capture the context of entire utterances, which is difficult to do with standard N-grams. [21, 22] have also shown that RNNLMs can significantly improve the performance of speech recognition, especially when RNN models are interpolated with N-gram language models. However, the drawback of RNNLMs is the computational complexity. Therefore, this type of language model is usually used for rescoring in two-pass decoding systems.

The systems that we developed for the IWSLT challenge adopt a class-based RNNLM [21], which consists of 1 hidden layer with 150 hidden nodes and 400 classes. The model is trained using the threaded version of the RNNLM toolkit. It took about 1 day to finish the training process.

2.5. Decoding strategy

For the test evaluation period we had 3 Gaussian mixture model hidden Markov model (GMM-HMM) systems and 12 DNN systems at hand for decoding that made use of 3 different front-ends. The GMM-HMM systems are trained using SAT. The DNNs use 3 different types of activation functions and 2 training criteria (see table 1). The GMM-HMM based SAT systems serve as basis for the sigmoid DNN systems, since their neural nets were built on top of the fMLLR transforms from these systems. We trained all systems on the same data, and they use the same lexicon and language models during decoding and rescoring. We run the decoding with a pruned 4-gram language model. Subsequent lattice rescorings make use of a 5-gram language model and an RNNLM language model. Given the lattices, we apply minimum Bayes risk (MBR) [23] decoding for all systems to minimize the expected word error rate (WER). After rescoring, we perform system combination using ROVER. To benefit from the individual system strengths, we attempted to apply a rank-score based weighting scheme that was first introduced in [7]. System weights during combination are conditioned to their respective rank-score. Let $\mathrm{rank}(s_n) \in \{1, \ldots, |\mathcal{S}|\}$ be the rank of a system $s_n \in \mathcal{S}$, where the system s^* with the highest accuracy $\mathrm{acc}(s^*)$ has rank 1. The rank-score of a system s_n is

$$\mathrm{acc}(s_n) \cdot (|\mathcal{S}| + 1 - \mathrm{rank}(s_n)) \tag{3}$$

A numerically lower rank indicates a system with higher performance. Weighting is performed according to:

$$\mathrm{weight}(s_n) = \frac{\mathrm{acc}(s_n) \cdot (|\mathcal{S}| + 1 - \mathrm{rank}(s_n))}{\sum_{s_n \in \mathcal{S}} \mathrm{acc}(s_n) \cdot (|\mathcal{S}| + 1 - \mathrm{rank}(s_n))} \tag{4}$$

Corpus	Amount
BN 1996	81.79 h
BN 1997	72.36 h
TED-LIUM	200.00 h
TIMIT	3.92 h
WSJ	81.01 h
Total	439.08 h

Table 2: *Training data for acoustic modeling.*

Corpus	Word count
EUROPAL	49.13 M
GIGA	567.76 M
NC	1.17 M
TED-LIUM	2.25 M
WSJ	36.98 K

Table 3: *Training data for language modeling.*

Since for ROVER implicit weighting according to Equation (4) was not possible, we used an approximate method where hypotheses are taken into consideration multiple times for the combination, according to their respective ranks: In a combination of 4 systems, the best system enters ROVER 4 times, the second best 3 times and so on.

3. Data resources

For the IWSLT 2015 evaluation, the regulations regarding the permissible training data are less restrictive, with no explicit cut-off date for data set. Data for language modeling is generally unrestricted, whereas acoustic modeling has to exclude a number of selected TED and TEDx talks that are not permitted to be used for training.

3.1. Acoustic model training data

For the ASR acoustic modeling no training data is provided, in contrast to the other evaluation tracks. Since data selection is unrestricted with the above mentioned exceptions, we were able to freely choose our speech corpora. The data we used for training acoustic models is selected from various resources including TED-LIUM corpus release 2 [24], Broadcast News [25], WSJ [26], and TIMIT [27], as listed in Table 2. We utilized TED-LIUM instead of the original downloadable TED talks because TED-LIUM is an already cleaned, noise-free corpus, and provides a good basis for training a full-fledged speech recognition system [24]. Although TIMIT is a relatively small corpus, it is suitable for training an initial monophone acoustic model.

3.2. Language model training data

The data for training language models comes from different sources including WSJ, EUROPARL, GIGA, NC, and TED, as shown in

Data	Amount
Speech (TED)	343 min
Noises (TED)	342 min
Noises (Soundsnap)	12 min
Total	697 min

Table 4: *Training data for the GMM segmenter training.*

Table 3. The data is cleaned by removing all punctuation, and removing case sensitivity by uppercasing all characters.

3.3. Evaluation data

With regards to the test corpora, the data set "tst2013" used in past editions as either an evaluation set (2013) or a progressive test set (2014) was provided by the organizers as the official development set for this year's evaluation. "tst2014" is used as a progressive test set, and a newly released test set "tst2015" consisting of 28 talks serves as the official test set for the final evaluation of all systems. Automatic segmentation of the raw audio data prior to decoding is a mandatory sub-task of the ASR track since 2013. We describe our approach for generating an automatic segmentation of the evaluation data in the following section.

4. Automatic segmentation of evaluation data

Given our observations regarding the effectiveness of neural net based and GMM based approaches for speech segmentation in previous work [7], we picked GMM-based segmentation as our method of choice for the IWSLT evaluation. This method uses a Viterbi decoder and GMM-HMM models to classify consecutively observed feature vectors into several sound categories. The mechanics of the general framework are comparable to the one presented in [28]. To improve segmentation quality, we experimented with data selection and model selection. We also tested the effectiveness of model combination to improve the final segmentation accuracy.

4.1. Segmentation training data

We used about 11.6 hours of data for model training, consisting of the official IWSLT "dev2010", "dev2012" and "tst2010" spoken utterances, noises extracted from the TED portion of the data used in [29, 30] and hand picked and manually trimmed noise samples downloaded from Soundsnap (`http://www.soundsnap.com`). Instead of keeping the detailed transcriptions, each spoken utterance in the test sets was labeled with a single *speech* token. A noise utterance is either labeled as *applause*, *laughter*, *music* or general *noise*. Table 4 lists the data for segmenter training.

4.2. Segmentation model

The general GMM segmentation framework is essentially a speech recognizer that is capable of discriminating several classes of sounds. Consecutive frames of the same sound are modeled as being generated by multi-state feed forward HMMs without skip states, where the minimal segment lengths are directly modeled by the HMM topology. Each GMM consists of 128 Gaussian components. The input is 42 dimensional LDA transformed MFCCs after stacking with a context of 7. The acoustic model is trained according to the maximum likelihood criterion, where the GMMs grow incrementally in several iterations of "split-and-merge" training [31]. The system is configurable by several parameters, one of which is a padding factor that expands hypothesized speech segments on both sides by a certain amount of milliseconds. This factor is tuned on the segmentation of this year's official development set. Segment coverage is computed on frame level and evaluated in terms of accuracy (ACC), true positive rate (TPR) and true negative rate (TNR).

Classes	Pad	ACC	TPR	TNR
[s],[sil+a+l]	0.325	88.9%	97.6%	45.6%
[s],[sil],[a+l]	0.475	90.1%	95.7%	62.2%
[s],[sil],[a],[l]	0.575	89.6%	95.8%	58.5%
[s],[sil],[a],[l],[n]	0.6	89.4%	95.9%	57.2%
[s],[sil],[a],[l],[n],[m]	0.8	82.6%	86.0%	65.7%

Table 5: *Segmenter performance dependent on the amount of classes. In column "Classes", the abbreviations stand for* speech, sil*ence,* a*pplause,* l*aughter, (general)* n*oise and* m*usic, respectively. Brackets delimit the individual classes formed by the data. Padding factors are in msec.*

Data (types)	Pad	ACC	TPR	TNR	WER
a+l+n+m	0.65	88.9%	95.5%	56.2%	27.3%
a+l+n	0.8	88.1%	95.3%	52.0%	28.8%
a+l	0.475	90.1%	95.7%	62.2%	26.5%
a	0.4	90.2%	96.1%	61.0%	26.0%
-	0.475	89.4%	96.5%	53.9%	26.7%
combined		90.4%	97.5%	55.2%	25.7%

Table 6: *Segmenter performance dependent on the amount of data. Padding factors are in msec.* combined *is the weighted combination of segmentations.*

4.3. Sound class selection

We evaluated the impact of the amount of target sound classes. The most simple system is discriminating speech from non-speech, the most complex system separates the distinct noises into individual classes. Silence in the speech recordings was detected via a simple power threshold during the sample extraction step of the training pipeline and where silence is a class of it's own, these features are used as samples for a *silence* class. Table 5 lists the details of the systems subject to comparison.

It is noteworthy that the 5 class and 6 class models were trained on more data, since they model additional classes for specific sounds. For the 2 class and 3 class models several noise types were simply mapped to one broad noise class. We interpret the results in the following way: It seems 2 classes are less suited to properly discriminate non-speech from speech, given the relatively low TNR, whereas 6 classes make significantly more errors in classifying speech correctly. The adding of samples for music obviously leads to a better noise classification, but also to more confusions in classification of speech. The 3 class model segmentation yields the highest accuracy, showing a comparatively good TNR with little loss in TPR given the alternatives. All further experiments were undergone with the 3 class segmentation model.

4.4. Sound class combination

We trained several models to test the impact of including or or excluding data of distinct noise types during training. The speech data and the hand picked Soundsnap samples were kept fix, and different portions of the TED noises were added. For each set generated this way, a segmenter was trained, tuned and evaluated. The results in table 6 show that it is the original data set that leads to

	Segmentation →	manual			automatic		
	Features →	MFCC	PLP	FBANK	MFCC	PLP	FBANK
Model	GMM-HMM (SAT)	23.9%	23.9%	24.8%	24.4%	24.8%	25.4%
	Sigmoid DNN (CE)	14.4%	-	-	15.1%	-	-
	ReLU DNN (CE)	11.2%	10.9%	12.7%	12.0%	11.7%	13.5%
	p-norm DNN (CE)	10.8%	10.5%	12.6%	11.4%	11.4%	13.5%
	p-norm DNN (sMBR)	**9.8%**	-	11.2%	**10.5%**	-	11.8%

Table 7: *Individual system performances of all recognizers in WER after rescoring.*

Systems								Weights	
ReLU DNN (CE)			p-norm DNN (CE)			p-norm DNN (sMBR)			
MFCC	PLP	FBANK	MFCC	PLP	FBANK	MFCC	FBANK	equal	rank-score
✓	✓	✓	✓	✓	✓	✓	✓	10.0%	9.7%
✓	✓		✓	✓	✓	✓	✓	9.8%	9.7%
✓	✓		✓	✓		✓	✓	9.8%	9.5%
	✓		✓	✓		✓	✓	9.6%	9.7%
			✓	✓		✓	✓	9.6%	9.7%
				✓		✓	✓	**9.5%**	9.6%
				✓		✓		10.0%	9.8%

Table 8: *Comparison of the 8 best ROVER combinations with equal and rank-score based weighting. Performance is measured in WER.*

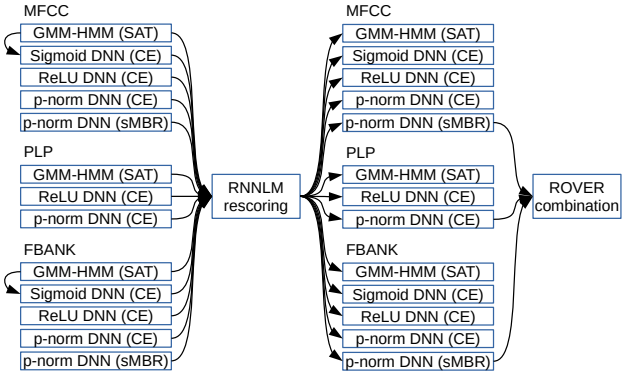


Figure 2: *Decoding pipeline of the primary submission. The leftmost arrows symbolize the dependency of the sigmoid DNNs on the fMLLR transforms of the GMM-HMM systems.*

optimal performance. If more noises are added, the performance deteriorates. If less noises are seen during training, the speech classification performance increases, while at the same time noise classification suffers. The table also lists the decoding performance of the SAT models, when decoded given the respective segmentations. The baseline performance on the provided segmentation is 25.0% WER. To benefit from the individual model strengths, we successfully applied the rank-score based weighting scheme of subsection 2.5 to combine segmentations on frame level. Since combination is performed frame-wise, artifacts in form of extremely short segments may be introduced at positions where the models greatly differ in their prognoses. To counter-act this phenomenon, segments are merged according to the heuristic

$$\text{from}(seg_2) - \text{to}(seg_1) \leq \delta \wedge \text{to}(seg_2) - \text{from}(seg_1) \leq \theta \quad (5)$$

with δ being subject to tuning (40 msec during our experiments) and θ set to 30000 msec. The weighted combination improves segmentation accuracy as well as speech recognition performance, reducing the WER to 25.7%. Combination with equal weights yielded similar results, but was inferior to our proposed method.

5. ASR evaluation

We evaluated our ASR systems on the "tst2013" development set, given the manual segmentation as well as our own, automatically generated segmentation. In preliminary experiments we found that RNNLM rescoring consistently outperformed rescoring with the 5-gram LM, producing WERs that were 0.4% absolute better on average. Thus, the results presented in this section only cover the results after RNNLM rescoring.

5.1. Single system performance

Table 7 lists the single system performances of all successful decodings on the development set. PLP features generally helped to achieve the best performance, followed by MFCC features. The gap between the MFCC and FBANK features is fairly large. It can also be seen that DNNs utilizing the p-norm activation function exceed the other nets' classification capabilities. Finally, the nets trained with the sMBR training criterion led to better accuracy than the ones built according to the cross-entropy criterion. The apparent inferiority of the sigmoid DNN might be due to several reasons, one of which is the differing activation function, given that ReLU seems to have an advantage on large data, according to previous work [11]. Another reason might be the differing network layout. Our assumption however is that the main difference is caused by the fact that this model is using standard features only, without the i-vectors stacked on top. This matches our observations in [7], where we used the same layout for all NNs and still observed a large gap between the system's performance. This thus re-confirms our assumption regarding the role of the features.

Decoding for the final submission had to be run on the automatic segmentation. Table 7 therefore also lists the recognition performance in WER for our own segmentation, created with the framework described in Section 4. Assuming that the scoring is identical or almost identical – given that we used the evaluation's default toolkit NIST SCTK (`http://www.nist.gov/itl/iad/mig/tools.cfm`) –

our single best system (p-norm DNN sMBR) already outperforms last year's winner in the ASR track by 0.1% absolute on "tst2013".

5.2. System combination performance

Table 8 lists the performance of weighted system combination using the rank-score function compared to the default combination with equal weighting of all systems. To guarantee that the systems are diverse enough to benefit from the combination, each combination of more than 2 systems covered all three front-ends. Experiments confirmed that failing to do so indeed leads to sub-optimal combinations that are not even able to beat the single best system.

The results are interesting insofar as it seems that improvement by weighting is not possible if the standard ROVER already leads to a better performance than the single best system involved in the combination. In cases where unweighted ROVER produces a sub-optimal result, weighting is able to boost the positive effects of combination and achieves a better result. This observation is consistent with the combination results of our segmentation in Subsection 4.4 as well as in [7]. Given the results on "tst2013" we performed the ROVER combination with equal weights on the automatically segmented set and achieved a WER of 10.1%. The system design of our primary submission is highlighted in Fig. 2.

6. Conclusion

This paper described the structure and development of NAIST's English ASR system for the English ASR track of the IWSLT 2015 evaluation campaign. We evaluated different architectures of deep neural network based models as well as various types of input features such as MFCC, PLP, FBANK and i-vector. The results show that a p-norm DNN trained on combined MFCC + i-vector feature vectors following the sMBR training criterion achieves the best performance for a single system, yielding a WER of 9.8% on the official development set. After system combination with ROVER, where the outputs of the best systems for each front-end were combined, the WER can be further reduced to 9.5%.

We trained several simple GMM models for speech/non-speech classification for the purpose of automatic segmentation prior to decoding. To exploit the benefits of multiple models we performed a rank-score based weighting in a segmentation hypothesis combination scheme on frame level. The combined segmentation outperforms the single best segmentation in terms of segment coverage accuracy and WER after actual decoding. Our best decoding on the automatically segmented development set achieves a 10.1% WER, which outperforms last year's winning system by 0.5% absolute WER on this set. This setup was used for producing our primary submission for the evaluation campaign. The official scoring of our primary submission on the "tst2015" evaluation set yields 12.0% WER.

7. Acknowledgements

Part of this research was supported by JSPS KAKENHI Grant Numbers 26870371 and 24240032, and the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan.

8. References

[1] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *Proceedings of ASRU*, 2001.

[2] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The Kaldi speech recognition toolkit," in *Proceedings of IEEE workshop*, 2011.

[3] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proceedings of ASRU*, Dec. 1997, pp. 347–354.

[4] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[5] B. Hoffmeister, D. Hillard, S. Hahn, R. Schlüter, M. Ostendorf, and H. Ney, "Cross-site and intra-site ASR system combination: Comparisons on lattice and 1-best methods." in *Proceedings of ICASSP*, 2007, pp. 1145–1148.

[6] K. Audhkhasi, A. M. Zavou, P. G. Georgiou, and S. Narayanan, "Empirical link between hypothesis diversity and fusion performance in an ensemble of automatic speech recognition systems." in *Proceedings of INTERSPEECH*, 2013, pp. 3082–3086.

[7] Q. T. Do, M. Heck, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "The NAIST ASR system for the 2015 multi-genre broadcast challenge: On combination of deep learning systems using a rank-score function," in *Proceedings of ASRU*, 2015.

[8] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, pp. 357–366, 1980.

[9] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[10] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front end factor analysis for speaker verification," *IEEE*, 2010.

[11] D. Povey and G. Saon, "Feature and model space speaker adaptation with full covariance gaussians." in *Proceedings of INTERSPEECH*, 2006.

[12] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proceedings of ICASSP*, May 2014, pp. 215–219.

[13] D. Povey and B. Kingsbury, "Evaluation of proposed modifications to MPE for large scale discriminative training," in *Proceedings of ICASSP*, vol. 4, April 2007, pp. IV–321–IV–324.

[14] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition." in *Proceedings of INTERSPEECH*, 2006.

[15] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[16] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proceedings of ICASSP*, 2013, pp. 8609–8613.

[17] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of deep neural networks with natural gradient and parameter averaging," *arXiv*, 2014.

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

[19] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of ACL*, 1996, pp. 310–318.

[20] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proceedings of ICSLP*, vol. 2, Denver, USA, 2002, pp. 901–904.

[21] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proceedings of ICASSP*, 2011, pp. 5528–5531.

[22] S. Kombrink, M. K. T. Mikolov, and L. Burget, "Recurrent neural network based language modeling in meeting recognition," in *Proceedings of INTERSPEECH*, 2011, pp. 2877–2880.

[23] 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.

[24] A. Rousseau, P. Delglise, and Y. Estve, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *Proceedings of LREC*, 2014.

[25] D. Graff, "The 1996 broadcast news speech and language-model corpus," in *Proceedings of DARPA Speech Recognition Workshop*, 1996, pp. 11–14.

[26] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proceedings of the DARPA SLS Workshop*, 1992, pp. 357–362.

[27] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and L. D. Nancy, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NICT*, vol. 93, p. 27403, 1993.

[28] M. Heck, S. Mohr, C. Stüker, M. Müller, 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. Železný, I. Habernal, and A. Ronzhin, Eds. Springer International Publishing, 2013, vol. 8113, pp. 286–293.

[29] C. Saam, C. Mohr, K. Kilgour, M. Heck, M. Sperber, K. Kubo, S. Stüker, S. Sakti, G. Neubig, T. Toda, S. Nakamura, and A. Waibel, "The 2012 KIT and KIT-NAIST English ASR systems for the IWSLT evaluation," in *Proceedings of IWSLT*, 2012.

[30] M. Heck, K. Kubo, M. Sperber, S. Sakti, S. Stüker, 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 IWSLT*, 2012.

[31] 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.

Improvement of Word Alignment Models for Vietnamese-to-English Translation

Takahiro Nomura, Hajime Tsukada, Tomoyoshi Akiba

Toyohashi University of Technology,
Aichi, Japan

nomura@nlp.cs.tut.ac.jp,tsukada@brain.tut.ac.jp,akiba@cs.tut.ac.jp

Abstract

Aiming at better SMT systems, two approaches for improving word alignment between Vietnamese and English are proposed and evaluated. One is to delete English words that never appear in Vietnamese; the other is to retokenize Vietnamese tokens so that each token of Vietnamese matches an English word. Although the baseline systems could not be improved by these methods at this moment, the results of the analysis show that these approaches are promising.

1. Introduction

Nowadays, a large number of bilingual corpora between popular languages such as English, Chinese, Arabic, and European languages (as listed in the "permissible training data" in this evaluation campaign) are available. In contrast, few corpora for many Asian languages are available. Although Vietnamese was one of the low resource languages, the TED task provided a fair amount of bilingual corpora between English and Vietnamese. Accordingly, Vietnamese has becomes a new target of statistical machine translation.

Since tokenization and grammatical constituents of Vietnamese are different from those of English, each token or word does not always correspond to an English word. This nature of Vietnamese leads a poor word alignment model between Vietnamese and English that will be a base of phrase alignment. To overcome this problem, two methods are proposed: (a) deleting English words that never appear in Vietnamese and inserting them afterward and (b) retokenize Vietnamese so that each token corresponds to an English word. To the authors' knowledge, this is the first application of these methods to Vietnamese translation. Although the baseline system could not be improved by these methods at this moment, we believe these methods will be helpful with further improvement.

The rest of the paper is organized as follows. Section 2 reviews the Vietnamese language. Section 3 explains the method used for retokenization and Section 4 explains our system configuration. Section 5 presents the results of an experimental evaluation of the proposed system, and Section 6 discusses the results. Section 7 concludes the paper.

2. Vietnamese language

Key features of the Vietnamese language are summarized as follows. Some Vietnamese sentences and their English translations are shown in Figure 1. The following features of the Vietnamese language can be seen in this example:

1. Vietnamese is tokenized into units that correspond approximately to syllables.

2. Vietnamese does not have words equivalent to English articles.

For example, in Figure 1, "kêt́ quá" corresponds to "result". Also, there is no Vietnamese word corresponding to the English article "the."

The English side of the training data of the experiment has 2,492,239 words and the number of the articles is 213,710. Therefore, approximately 9% of the English words do not correspond to Vietnamese words. Since the Vietnamese side of the training data has 3,030,127 words, the Vietnamese sentence is approximately 1.3 times longer than that of English ignoring English articles that do not correspond to Vietnamese words.

To improve word alignment models, it is preferable to retokenize Vietnamese words into a unit that corresponds to an English word. Also, an English article must be deleted from the viewpoint of word alignment models, if possible. Considering the former point, we apply two tokenization methods (explained in Section 3). Considering the latter point, we apply two-step translation (explained in Section 4).

3. Tokenizer

Hereafter, the term "tokenization" is simply used for "retokenization" of Vietnamese words where some original tokens are consolidated.

Two tokenization systems are used. One is an existing tokenization tool for Vietnamese, namely, vnTokenizer[1]. It utilizes a word dictionary. Therefore, we refer this as a supervised method. The other is an unsupervised tokenizer proposed by Tagyoung et al. [2] and does not utilize any word dictionaries.

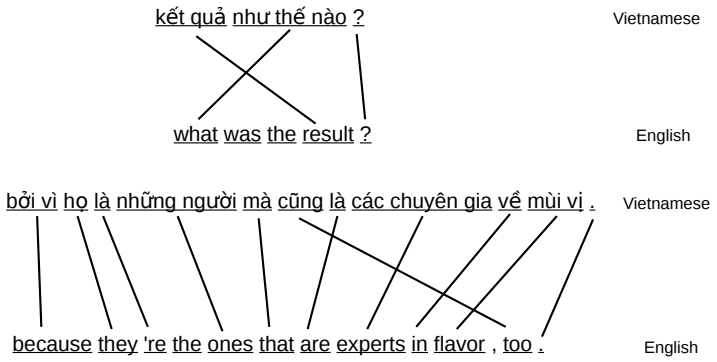


Figure 1: Phrase alignment of Vietnamese and English

3.1. Unsupervised bilingual tokenizer

This tokenization method based on a word-level alignment model trained by using a parallel corpus was originally proposed by Tagyoung et al. for languages that are not tokenized by spaces (such as Chinese and Korean). It is used here for consolidating original Vietnamese tokens.

3.1.1. Bilingual model

The bilingual model is denoted by the following equation. The input data are a tokenized English string e^n and an untokenized Vietnamese string s^m, where "untokenized" means the original tokens are left.

$$P(f, a = k|e) = \frac{\alpha(i)P(f_i|e_k)P(a = k)\beta(j)}{P(\mathbf{c}|\mathbf{e})}$$

where $f = \{s_i s_{i+1} ... s_j\}$ is a new token formed by concatenating from the i-th to the j-th Vietnamese tokens, and a is a variable indicating the position of the English word that generates f. α and β are given by the following equations:

$$\alpha(i) = \sum_{l=1}^{L} \alpha(i-l) \sum_{a} P(a)P(s_{i-l}^{i}|e_a)$$

$$\beta(j) = \sum_{l=1}^{L} \sum_{a} P(a)P(s_{j}^{j+l}|e_a)\beta(j+l)$$

where L is the maximum syllable length for a word.

This model is trained by using an EM algorithm. First, it calculates the expected counts of individual word pairs:

$$ec(s_i^j, e_k) = \frac{\alpha(i)P(a)P(s_i^j|e_k)\beta(j)}{\alpha(m)}$$

Second, an M step simply normalizes the counts:

$$P(f|e) = \frac{ec(f, e)}{\sum_f ec(f, e)}$$

Given two sentences, **e** and **f**, the optimal segmentation of a new source-language sentence can be obtained by using the Viterbi algorithm.

$$segments = argmin_{\mathbf{s}} \sum_{i}^{n} \left(-log \sum_{\mathbf{a}} P(s_i|e_a) + \theta \right)$$

where $\mathbf{s} = \{s_1 s_2 ... s_n\}$ is a segment set of source sentences **f**, and **a** is the alignment of the source segments to the target words. This model can be applied only when a target sentence is available.

3.2. Monolingual model

The monolingual model is denoted by the following equation.

$$P(f) = \sum_{e} P(f|e)P(e)$$

where $P(f|e)$ is the probability of the bilingual model explained Section 3.1.1. $P(e)$ is a monolingual model calculated by the following equation.

$$P(e_i) = \frac{count(e_i)}{\sum_k^K count(e_k)}$$

where $count()$ is the number of occurrences on the English side of the training corpus, and K is the size of the vocabulary.

4. System configuration

The configuration of the proposed system is shown in Figure 2. Each SMT system, namely, (a) baseline system,(b) two-step translation system, and (c) retokenized system, preform the translation for the test set. Multi-Engine Machine Translation (MEMT) then performs the system combination. It receives the results of the combined systems as inputs.

4.1. Baseline system

A phrase-based SMT system and a hierarchical phrase-based SMT system were adopted as baseline systems. These systems are trained by Moses scripts from parallel corpus that is tokenized. The phrase table is trained by the grow-diag-final method and the reordering model is msd-bidirectional-fe.

4.2. Two-step translation system for inserting articles

Two-step translation was performed to deal with English articles properly. The first step is a translation from Vietnamese to English that erases the article. The second step is a translation from English without articles to original English.

First, this two-step approach makes a corpus in which articles of the English side of the parallel corpus are removed and to makes a trilingual parallel corpus: both languages of the original parallel corpus and the newly made English corpus without articles. The two systems are trained by using the trilingual corpus. The first system is a phrase-based SMT system or hierarchical phrase-based SMT system trained by Vietnamese and English without articles. This system receives Vietnamese as an input and outputs English without articles. The second system is a phrase-based system trained by English without articles and original English. It receives the first system's output as an input and complements the removed articles.

4.3. Retokenized system

The training set is tokenized by using vnTokenizer and unsupervised Tokenizer. A phrase-based SMT system is trained by using these tokenized corpora.

The phrase-based systems are trained by a corpus tokenized by vnTokenizer and unsupervised Tokenizer. The systems may have more unknown words than the baseline system because retokenization may not be consistent and produce unknown combined tokens. To solve this problem, the tokens in the phrase table are divided, and the original notation is recovered after the phrase table is built. This system does not perform two-step translation in the experiment.

4.4. System combination

To improve of the translation quality, the outputs of each system are combined by using MEMT[3] (developed by Kenneth Heafield et al).

5. Experiment

The effectiveness of our proposed methods was experimentally evaluated by using a Vietnamese-to-English translation task in IWSLT2015.

5.1. Submitted results

Our submitted results used all of the development sets and test sets provided by IWSLT2015 as a development set. Contrastive1 is the result given by the hierarchical phrase-based baseline system. Contrastive2 is the result given by the phrase-base baseline system. Contrastive3 is the result given by the hierarchical two-step translation system. Contrastive4 is the result given by the phrase-based two-step translation system. Contrastive5 is the result given by the phrase-based retokenized system. The primary is the results obtained by MEMT combining all results listed above. However, the systems had some bugs. The following shows the results where the bugs were fixed.

5.2. Conditions

Only in-domain training and development data of TED talks provided for the IWSLT2015 evaluation campaign were used in the experiments.

Both languages in the training set were tokenized, and the first letter of sentences was recased. The case of the original form is determined by the majority in the training data. The training data was cleaned so that the length of a sentence is 80 words at most.

A language model was created by using test set IWSLT2015 to select the development set based on the perplexity, and test set IWSLT2010 was adopted as the development set.

Moses[4] was used for translation tools, and GIZA++[5] for word alignment tools. The language model was trained by using kenLM[6], and MEMT was used for combining the systems.

5.3. Results

The results obtained by the proposed system are listed in Table 1. In this table, the baseline denotes the systems explained in Section 4.1, and "two-step translation" is the system explained in Section 4.2. The method "vnTokenizer" utilized the corpus tokenized by vnTokenizer. The method "unspTokenizer(bi)" utilized the corpus tokenized by the bilingual model described in Section 3.1.1, and the method "unspTokenizer(mono)" utilized the corpus tokenized by the monolingual model described in Section 3.1.2. And "system comb" is the SMT system that combines all systems listed above.

The two-step translation for articles and retokenization of Vietnamese could not improve the performance of the baseline systems. Also, the result of the system combination fell below the baselines.

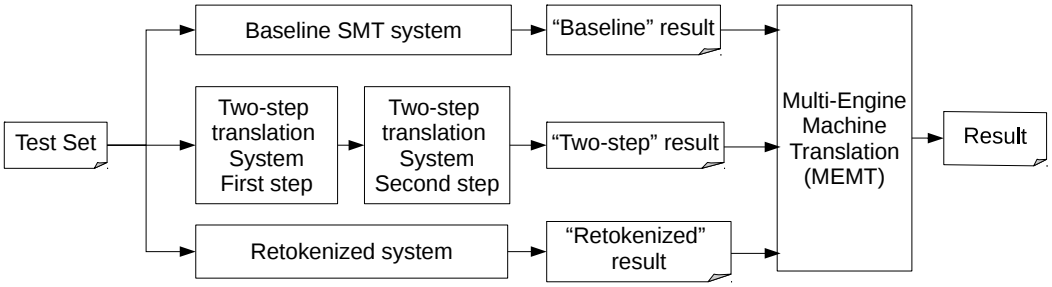


Figure 2: Outline of proposed system

method	model	BLEU %
baseline	phrase base	24.41
	hierarchical	25.00
two-step translation	phrase base	19.06
	hierarchical	19.22
retokenized	vnTokenizer+phrase base	20.38
	unspTokenizer(bi)+phrase base	19.11
	unspTokenizer(mono)+phrase base	19.97
system comb		20.78

Table 1: Experiment results

6. Discussion

As for the two-step translation, the performance improvement of the first step is worse than we expected. The BLEU score of the first step in the development set is 23.47 and that of the baseline system ignoring English articles is 23.34. We have no idea on the very small improvement at this moment. Clearly, this problem must be further investigated.

As for retokenization, vnTokenize may cause mismatch between the training data and the TED task, and its tokenization performance may not be good enough. The unsupervised tokenizer does not cause task mismatch between training data and test data. However, the model does not guarantee each tokenized unit corresponds to an English word, although the model considers bilingual natures. This is a weakness of the current model of the unsupervised tokenizer.

In addition, the result given by the unsupervised tokenizer is not consistent. Therefore, it causes a large number of out-of-vocabulary words if the phrase table is used without reforming the original tokens.

7. Conclusion

Two methods for improving baseline translation were applied. One is deleting English articles that never appear in Vietnamese and inserting them afterward. The other is to retokenize Vietnamese so that each Vietnamese word corresponds to an English word by applying both supervised and unsupervised tokenizers. Although these methods were not helpful at the moment, our analysis shows that the approaches themselves are promising.

8. Acknowledgment

The advice and comments given by Doan Thi Thuy Trinh greatly helped us to understand the Vietnamese language.

9. References

[1] L. H. Phuong, N. Thi Minh Huyên, A. Roussanaly, and H. T. Vinh, "Language and automata theory and applications," C. Martín-Vide, F. Otto, and H. Fernau, Eds. Berlin, Heidelberg: Springer-Verlag, 2008, ch. A Hybrid Approach to Word Segmentation of Vietnamese Texts, pp. 240–249. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88282-4_23

[2] T. Chung and D. Gildea, "Unsupervised tokenization for machine translation," in *In Proc. EMNLP 2009*, 2009.

[3] K. Heafield and A. Lavie, "Combining machine translation output with open source: The Carnegie Mellon multi-engine machine translation scheme," *The Prague Bulletin of Mathematical Linguistics*, vol. 93, pp. 27–36, January 2010. [Online]. Available: http://kheafield.com/professional/avenue/marathon2010.pdf

[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 ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 177–180. [Online]. Available: http://dl.acm.org/citation.cfm?id=1557769.1557821

[5] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[6] 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

Technical Papers

Unsupervised comparable corpora preparation and exploration for bi-lingual translation equivalents

Krzysztof Wołk, Krzysztof Marasek

Department of Multimedia
Polish-Japanese Academy of Information Technology, Koszykowa 86, 02-008 Warsaw
kwolk@pja.edu.pl, kmarasek@pja.edu.pl

Abstract

The multilingual nature of the world makes translation a crucial requirement today. Parallel dictionaries constructed by humans are a widely-available resource, but they are limited and do not provide enough coverage for good quality translation purposes, due to out-of-vocabulary words and neologisms. This motivates the use of statistical translation systems, which are unfortunately dependent on the quantity and quality of training data. Such systems have a very limited availability especially for some languages and very narrow text domains. In this research we present our improvements to current comparable corpora mining methodologies by re-implementation of the comparison algorithms (using Needleman-Wunch algorithm), introduction of a tuning script and computation time improvement by GPU acceleration. Experiments are carried out on bilingual data extracted from the Wikipedia, on various domains. For the Wikipedia itself, additional cross-lingual comparison heuristics were introduced. The modifications made a positive impact on the quality and quantity of mined data and on the translation quality.

1. Introduction

The aim of this research is a preparation of parallel and comparable corpora and language models. This work improves SMT quality through the processing and filtering of parallel corpora and through extraction of additional parallel data from the resulting comparable corpora. To enrich the language resources of SMT systems, adaptation and interpolation techniques have been applied to the prepared data. Experiments were conducted using data from a wide domain (TED[1] presentations on various topics).

Evaluation of SMT systems was performed on random samples of parallel data using automated algorithms (BLEU metric) to evaluate the quality and potential usability of the SMT systems' output [1].

As far as experiments are concerned, the Moses Statistical Machine Translation Toolkit software [2] is used. Moreover, the multi-threaded implementation of the GIZA++ tool is employed to train models on parallel data and to perform their symmetrization (using Berkeley Aligner [28]) at the phrase level. The statistical language models from single-language data are trained and smoothed using the SRI Language Modeling toolkit (SRILM). In addition, data from outside the thematic domain is adapted. In the case of parallel models, Moore-Lewis Filtering [3] is used for pseudo in-domain data selection, while single-language models are linearly interpolated [4].

Lastly, methodology proposed in the Yalign [5] parallel data mining tool is analyzed and enhanced. Its speed is increased by reimplementing it in a multi-threaded manner and by employing graphics processing unit (GPU) for its calculations. Quality is improved by using the Needleman-Wunch [6] algorithm for sequence comparison and by developing a tuning script that adjusts mining parameters to specific domain requirements.

The resulting systems out-performed baseline systems used in the tests.

2. Corpora Types

A corpus is a large collection of texts, stored on a computer. Text collections are called corpora. The term "parallel corpus" is typically used in linguistic circles to refer to texts that are translations of each other. For statistical machine translation, we are especially interested in parallel corpora, which are texts paired with a translation into another language. Preparing parallel texts for the purpose of statistical machine translation may require crawling the web, extracting the text from formats such as HTML, and performing document and sentence alignment [4].

There are two main types of parallel corpora, which contain texts in two languages. In a comparable corpus, the texts are of the same kind and cover the same content. An example is a corpus of articles about football from English and Polish newspapers. In a translation corpus, the texts in one language (e) are translations of texts in the second language (f). It is important to remember that the term "comparable corpora" refers to texts in two languages that are similar in content, but are not translations of each other [4].

To exploit a parallel text, some kind of text alignment, which identifies equivalent text segments (approximately, sentences), is a prerequisite for analysis.

Machine translation algorithms for translating between a first language and a second language are often trained using parallel fragments, comprising a first language corpus and a second language corpus, which is an element-for-element translation of the first language corpus. Such training may involve large training sets that may be extracted from large bodies of similar sources, such as databases of news articles written in the first and second languages describing similar events. However, extracted fragments may be comparatively "noisy", with extra elements inserted in each corpus. Extraction techniques may be devised that can differentiate between "bilingual" elements represented in both corpora and "monolingual" elements represented in only one corpus, and for extracting cleaner parallel fragments of bilingual elements. Such techniques may involve conditional probability determinations on one corpus with respect to the other corpus,

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

or joint probability determinations that concurrently evaluate both corpora for bilingual elements [4].

Because of such difficulties, high-quality parallel data is difficult to obtain, especially for less popular languages. Comparable corpora are the answer to the problem of lack of data for the translation systems for under-resourced languages and subject domains. It may be possible to use comparable corpora to directly obtain knowledge for translation purposes. Such data is also a valuable source of information for other cross-lingual, information-dependent tasks. Unfortunately, such data is quite rare, especially for the Polish–English language pair. On the other hand, monolingual data for those languages is accessible in far greater quantities [4].

Summing up, four main corpora types can be distinguished. Most rare parallel corpora can be defined as corpora that contain translations of the same document into two or more languages. Such data should be aligned, at least at the sentence level. A noisy parallel corpus contains bilingual sentences that are not perfectly aligned or have poor quality translations. Nevertheless, mostly bilingual translations of a specific document should be present in it. A comparable corpus is built from non-sentence-aligned and untranslated bilingual documents, but the documents should be topic-aligned. A quasi-comparable corpus includes very heterogeneous and non-parallel bilingual documents that may or may not be topic-aligned [18].

3. State of the art

As far as comparable corpora are concerned, many attempts (especially for Wikipedia) have been made so far to extract parallel data samples. Two main approaches for building comparable corpora can be distinguished. Perhaps the most common approach is based on the retrieval of cross-lingual information. In the second approach, source documents must be translated using any machine translation system. The documents translated in that process are then compared with documents written in the target language, to find the most similar document pairs.

An interesting idea for mining parallel data from Wikipedia was described in [8]. The authors propose two separate approaches. The first idea is to use an online machine translation (MT) system to translate Dutch Wikipedia pages into English, and then try to compare original EN pages with translated ones. The idea, although interesting, seems computationally infeasible, and it presents a chicken-egg problem. Their second approach uses a dictionary generated from Wikipedia titles and hyperlinks shared between documents. Unfortunately, the second method was reported to return numerous, noisy sentence pairs. The second method was improved in [9] by additional restrictions on the length of the correspondence between chunks of text and by introducing an additional similarity measure. They prove that [8] the precision (understood as number of correct translations pairs over total number of candidates) is about 21%, and in the improved method [9], the precision is about 43%.

Yasuda and Sumita [11] proposed an MT bootstrapping framework based on statistics that generate a sentence-aligned corpus. Sentence alignment is achieved using a bilingual lexicon that is automatically updated by the aligned sentences. Their solution uses a corpus that has already been aligned for initial training. They showed that 10% of Japanese Wikipedia sentences have an equivalent on English Wikipedia.

Interwiki links were leveraged by the approach of Tyers and Pienaar in [10]. Based on Wikipedia link structure, a bilingual dictionary is extracted. In their work, they measured the average mismatch between linked Wikipedia pages for different languages. They found that precision of their method is about 69-92% depending on language.

In [12] the authors attempt to advance the state of the art in parallel data mining by modeling document-level alignment using the observation that parallel sentences can most likely be found in close proximity. They also use annotation available on Wikipedia and an automatically-induced lexicon model. The authors report 90% recall and 80% precision.

The author of [13] introduces an automatic alignment method for parallel text fragments that uses a textual entailment technique and a phrase-based SMT system. The author states that significant improvements in SMT quality were obtained (BLEU increased by 1.73) by using this aligned data between German and French languages.

Another approach for exploring Wikipedia was recently described in [14] by M. Plamada and M. Volk. Their solution differs from the previously described methods in which the parallel data was restricted by the monotonicity constraint of the alignment algorithm used for matching candidate sentences. Their algorithm ignores the position of a candidate in the text and, instead, ranks candidates by means of customized metrics that combine different similarity criteria. In addition, the authors limit the mining process to a specific domain and analyze the semantic equivalency of extracted pairs. The mining precision in their work is 39% for parallel sentences and 26% for noisy-parallel sentences, with the remaining sentences misaligned. They also report an improvement of 0.5 points in the BLEU metric for out-of-domain data, and almost no improvement for in-domain data.

The authors in [15] propose obtaining only title and some meta-information, such as publication date and time for each document, instead of its full contents, to reduce the cost of building the comparable corpora. The cosine similarity of the titles' term frequency vectors were used to match titles and the contents of matched pairs.

In the research described in [16], the authors introduce a document similarity measure that is based on events. To count the values of this metric, they model documents as sets of events. These events are temporal and geographical expressions found in the documents. Target documents are ranked based on temporal and geographical hierarchies.

The authors in [17] also suggest an automatic technique for building a comparable corpus from the web using news web pages, Wikipedia, and Twitter in. They extract entities, time interval filtering, URLs of web pages, and document lengths as features for classification and for gathering the comparable data.

In the present research, a method inspired by the Yalign tool is used. The solution was far from perfect, but after improvements that were made during this research, it supplied the SMT systems with bi-sentences of good quality in a reasonable amount of time.

4. Parallel data mining

In this research, methodologies that obtain parallel corpora from data sources that are not sentence-aligned, such as noisy parallel or comparable corpora, are presented. The results of initial experiments on text samples obtained from Wikipedia

pages are presented. We chose Wikipedia as a data source because of the large number of documents that it provides (4,524,017 on EN wiki, at the time of writing). Furthermore, Wikipedia contains not only comparable documents, but also some documents that are translations of each other. The quality of the approach used was measured by improvements in MT systems translations.

For the experiments in data mining, the TED corpora prepared for the IWSLT 2015 evaluation campaign by FBK[1] were chosen. This domain is very wide and covers many unrelated subject areas. The data contains almost 2.5M untokenized words [19]. The experiments were conducted on DE-EN, FR-EN, VI-EN and CS-EN corpora.

The solution can be divided into three main steps. First, the comparable data is collected, then it is aligned at the article level, and finally the aligned results are mined for parallel sentences. The last two steps are not trivial, because there are great disparities between Wikipedia documents. This is most likely why sentences in the raw Wiki corpus are mostly misaligned, with translation lines whose placement does not correspond to any text lines in the source language. Moreover, some sentences have no corresponding translations in the corpus at all. The corpus might also contain poor or indirect translations, making alignment difficult. Thus, alignment is crucial for accuracy. Sentence alignment must also be computationally feasible to be of practical use in various applications.

Before a mining tool processes the data, texts must be prepared. Firstly, all the data is saved in a relational database. Secondly, our tool aligns article pairs and removes from the database articles that appear only in one of the two languages. These topic-aligned articles are filtered to remove any HTML tags, XML tags, or noisy data (tables, references, figures, etc.). Finally, bilingual documents are tagged with a unique ID as a topic-aligned, comparable corpus. To extract the parallel sentence pairs, a decision was made to try strategy designed to automate the parallel text mining process by finding sentences that are close translation matches from comparable corpora. This presents opportunities for harvesting parallel corpora from sources, like translated documents and the web, that are not limited to a particular language pair. However, alignment models for two selected languages must first be created.

The solution 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. It also uses a sequence aligner, which produces an alignment that maximizes the sum of the individual (per sentence pair) similarities between two documents [5].

For sequence alignment, the Yalign used an A* search approach [7] to find an optimal alignment between the sentences in two selected documents. The algorithm has a polynomial time worst-case complexity, and it produces an optimal alignment. Unfortunately, it cannot handle alignments that cross each other or alignments from two sentences into a single one [7].

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 is used: if the sentence similarity score is below it, the pair is excluded.

For the sentence similarity metric, the algorithm uses a statistical classifier's likelihood output and normalizes it into the 0–1 range.

The classifier must be trained in order to determine if sentence pairs are translations of each other. A Support Vector Machine (SVM) classifier was used in this research. Besides being an excellent classifier, an SVM can provide a distance to the separation hyperplane during classification, and this distance can be easily modified using a Sigmoid Function to return a value similar to likelihood between 0 and 1 [21].

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, good quality parallel data were needed, as well as a dictionary that included translation probability. For this purpose, we used the TED talks [18] corpora. To obtain a dictionary, we trained a phrase table and extracted 1-grams from it [22].

5. Improvements to the mining process

Unfortunately, the native Yalign tool was not computationally feasible for large-scale parallel data mining. The standard implementation accepts plain text or web links, which need to be accepted, as input, and the classifier is loaded into memory for each pair alignment. In addition, the Yalign software is single-threaded. To make the process faster, a solution was developed that supplies the classifier with articles from the database within one session, with no need to reload the classifier each time. The developed solution also facilitated multi-threading and decreased the mining time by a factor of 6.1x (using a 4-core, 8-thread i7 CPU). The alignment algorithm was also reimplemented for better accuracy and to leverage the power of GPUs for additional computing requirements. The tuning algorithm was implemented as well.

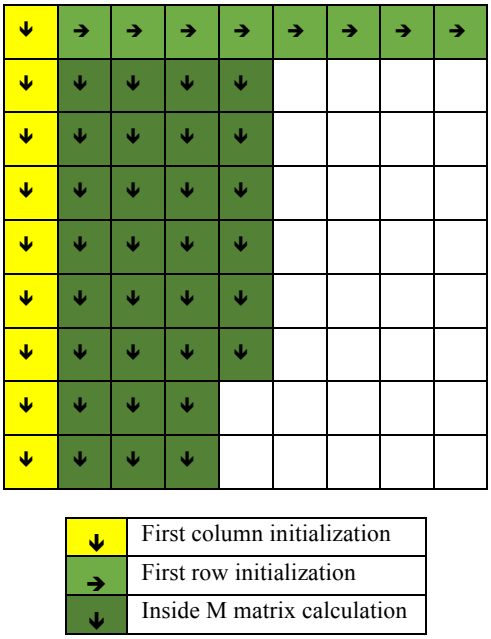
5.1. Needleman-Wunsch algorithm (NW)

The objective of this algorithm is to align two sequences of elements (letters, words, phrases, etc.). The first step consists of defining the similarity between two elements. This is defined by the similarity matrix S, an NxM matrix, where N is the number of elements in the first sequence and M is the number of elements in the second sequence. The algorithm originated in the field of bioinformatics for RNA and DNA comparison. However, it can be adapted for text comparison. In simple terms, the algorithm associates a real number with each pair of elements in the matrix. The higher the number, the more similar the two elements are. For example, imagine that we have the similarity matrix S (phrase-polish, phrase-english) = number between 0 and 1. A 0 for two phrases means they have nothing in common; 1 means that those two phrases are the exact translation of each other. The similarity matrix definition is fundamental to the results of the algorithm [7].

The second step is the definition of the gap penalty. It is necessary in the case when one element of a sequence must be associated with a gap in the other sequence; however, such a step will incur a penalty (p).

The calculation of the S matrix is performed starting from the S(0,0) element that is, by definition, equal to 0. After the first row and columns are initialized, the algorithm iterates through the other elements of the S matrix, starting from the upper-left side to the bottom-right side. Each step of this calculation is shown in Figure 1.

[1] http://www.fbk.eu/

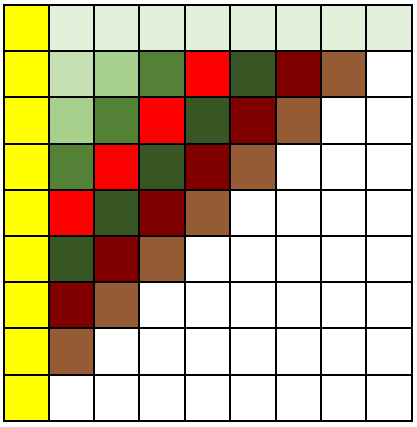


↓	First column initialization
→	First row initialization
↓	Inside M matrix calculation

Figure 1: Needleman-Wunsch S-matrix calculation

The two NW algorithms, with and without GPU optimization, are conceptually identical, but the first has an advantage in efficiency, depending on the hardware, of up to max(n, m) times.

It differs in the calculation of the S matrix elements. This calculation is the step to which multi-threading optimization is applied. Those operations are small enough to be processed by an enormous number of Graphics Processing Units (ex. CUDA cores). The idea is to compute all elements in a diagonal in parallel, always starting from the lower-left and proceeding to the bottom-right. An example is presented in Figure 2 [24].



	First column initialization in parallel threads
	First raw initialization in parallel
	1st diagonal calculation in parallel threads
	2nd diagonal calculation in parallel threads
	3rd diagonal calculation in parallel threads
	…

Figure 2: Needleman-Wunsch S-matrix calculation with parallel threads

The S matrix calculation starts from the top left column. In order to find out the value of a cell of S(m,n), for all pairs of m and n, the values to its top S(m-1,n) , left S(m,n-1) and top left S(m-1,n-1) must be known in advance. Where, S(m,n) can be calculated with the help of following equation, [26]:

$$S(m,n) = \max \{S(m-1) \pm 1,\ S(m-1,n) - 2,\ S(m,n-1) - 2\} \quad (1)$$

Nonetheless, the results of the A* algorithm, if the similarity calculation and the gap penalty are defined as in the NW algorithm, will be the same only if there is an additional constraint on paths: paths cannot go upward or leftward in the S matrix. Yalign does not impose these additional conditions, so in some scenarios, repetitions of the same phrase may appear. In fact, every time the algorithm decides to move up or left, it is coming back into the second and first sequence respectively.

An example of an S matrix without constraints is presented in Figure 3:

	a	d	e	g	f
a	X				
d		X			
c	X				
d		X			
e			X	X	X

Figure 3: S matrix pass trough without constraints

The alignment result in this scenario is:
a, d, a, d, e, g, f
a, d, c, d, e, −, −

In the same problem, the NW would react as presented in Figure 4:

	a	d	e	g	f
a	X				
d		X			
c		X			
d		X			
e			X	X	X

Figure 4: S matrix pass trough with NW

The alignment result using NW would be:
a, d, −, −, e, g, f
a, d, c, d, e, −, −

In order to visualize the problem, let us assume that first sequence is “tablets make children very addicted” and second one is “tablets make people spoil children”. The solution to

this sequence using A* algorithm without constrains is presented in Figure 5 and using NW in Figure 6.

	tablets	make	children	very	addicted
tablets	X	-	-	-	-
make	-	X	-	-	-
people	X	-	-	-	-
spoil	-	-	-	-	-
children	-	-	X	X	X

*Figure 5:A** alignment without constraints

	tablets	make	children	very	addicted
tablets	X	-	-	-	-
make	-	X	-	-	-
people	-	-	-	-	-
spoil	-	-	-	-	-
children	-	-	X	-	-

Figure 6: NW alignment with constraints

Because of the lack of constraints, repetitions were created that visualized the imperfection of the A* algorithm implemented in the Yalign program. Using A* many sentences may be misaligned or missed during the alignment, especially when analyzed texts are of different lengths and have vocabularies rich in synonyms. Some sentences can simply be skipped while checking for alignment. That is why NW with GPU optimization is more suitable algorithm. In this research, a comparison was made using all three approaches described above.

5.2. Tuning algorithm for classifier

The quality of alignments is defined by a tradeoff between precision and recall. The classifier has two configurable variables [25]:

- threshold: the confidence threshold to accept an alignment as "good." A lower value means more precision and less recall. The "confidence" is a probability estimated from a support vector machine classifying "is a translation" or "is not a translation." [27]
- penalty: controls the amount of "skipping ahead" allowed in the alignment [5]. Say you are aligning subtitles, where there are few or no extra paragraphs and the alignment should be more or less one-to-one; then the penalty should be high. If you are aligning things that are moderately good translations of each other, where there are some extra paragraphs for each language, then the penalty should be lower.

Both of these parameters are selected automatically during training but they can be manually adjusted if necessary. The solution implemented in this research also introduces a tuning algorithm for those parameters, which allows for better adjustment of them.

To perform tuning, it is necessary to extract random article samples from the corpus. Such articles must be manually aligned by humans. Based on such information, the tuning script tries, naively by random parameter selection, to find values for which classifier output is as similar to that of a human as possible. Similarity is a percentage value of how the automatically-aligned file resembles the human-aligned one. A Needleman-Wunsch algorithm is used for this comparison. Analysis was performed for each of four languages of interest to check how the tuning algorithms cope with proper adjustment of the parameters. Table 1 shows the results of this experiment. For testing purposes, 100 random article pairs were taken from the Wikipedia comparable corpus and aligned by a human translator. Second, a tuning script was run using classifiers trained on the previously described text domain. A percentage change in quantity of recognized parallel sentences was calculated for each classifier.

Table 1: Improvements in mining using tuning script for Wikipedia data

Domain	Improvement in %
DE	11.2
FR	13.5
CS	12.1
VI	15.2

5.3. Minor improvements for better Wikipedia exploration

All the improvements discussed in the previous sections deal mostly with heuristics used in the mining tool and can be applied to any bilingual textual data. Such an attitude would also improve the mining within Wikipedia. However, Wikipedia has many additional sources of cross-lingual dependencies that can be used. First of all, the topic domain of Wikipedia cannot be closed into a specific domain; this page covers almost any topic. This is the reason why its articles often contain complicated or rare vocabulary, and why statistical mining methods may skip some parallel sentences. The solution to this problem might be extraction of the dictionary using the article titles from Wikipedia (Figure 7) and additionally implemented web crawler tool.

Figure 7: Example of bi-lingual Wikipedia page title

According to [11] precision of such a dictionary can be as high as 92%. Such a dictionary can be used not only for the extension of the parallel corpora but also in the classifier training phase.

Secondly, figures, or to be more precise, their descriptions, contain good quality parallel phrases. It is possible to get them by picture analysis and hyperlinks to the pictures. The same goes for any figures, tables, maps, audio, video or any other multimedia contents on Wikipedia. Unfortunately, not all information can be extracted from Wikipedia dumps and it is required to use a web crawler suited for this task (Figure 8). This also means that only cross-lingual information that are annotated with common links can be extracted.

Figure 8: Example of bi-lingual figure caption

The good quality Wikipedia articles are well referenced. It is most likely for sentences to be cross-lingual equivalents if they are referenced with the same publication. Such analysis, joined with other comparison techniques, can lead to better accuracy in parallel text recognition (Figure 9).

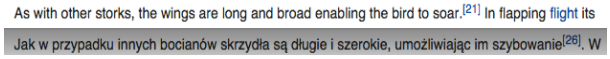


Figure 9: Example of bilingually referenced sentence

Unfortunately, the Wikipedia articles are developed separately for each language by many authors. In the following example, the parallel sentence in one language is notated with reference number 21 and in other with reference 26. It is why it is required not only to compare the number but to analyze the references themselves (Figure 10).

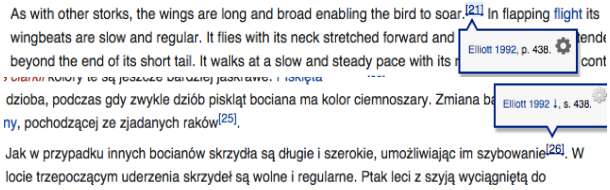


Figure 10. Example of cross-lingual reference

In addition to references, it is important to analyze names, dates, numbers etc. as well, because it they indicate parallel data presence.

Because of the need to use a web crawler, this tool version was evaluated only using 1,000 randomly selected articles. It would take too long to build using an entire corpus without access to many proxy servers and Internet connections. It is not only required to crawl about 5,000,000 articles for EN wiki but also many links present in each language Wiki.

Table 2: Number of parallel segments found

Y	4,192
YMOD	5,289
DICT	868
DICTC	685

Firstly, the data was crawled and secondly aligned using the standard version of the classifier (Y in Table 2) and then aligned using the modified version - dictionary, captions and references extraction as described above (YMOD in Table 2). Lastly, the single words were extracted and counted (DICT in Table 2) and also manually analyzed in order to verify how many of them could be considered as correct translations (DICTC in Table 2).

The results mean that the improved method using additional information sources mined an additional 1,097 parallel segments. Out of them, it possible to identify 868 single words, which means that in fact 229 new sentences were obtained. Potential growth in obtained data was equal to 5.5%. After manual analysis of the dictionary, 685 words were identified as proper translations. This means that the accuracy of the dictionary was about 79%.

5.4. Evaluation of improvements

As mentioned, some methods for improving the performance of the native classifier were developed. First, speed improvements were made by introducing multi-threading to the algorithm, using a database instead of plain text files or Internet links, and using GPU acceleration in sequence comparison. More importantly, two improvements were obtained to the quality and quantity of the mined data. The A* search algorithm was modified to use Needleman-Wunch, and a tuning script of mining parameters was developed. In this section, the CS-EN TED corpus will be used to demonstrate the impact of the improvements (it was the only classifier used in the mining phase). The data mining approaches used were: directional (CS->EN classifier) mining (MONO), bi-directional (additional EN->CS classifier) mining (BI), bi-directional mining using a GPU-accelerated version of the Needleman-Wunch algorithm (NW), and mining using the NW version of the classifier that was tuned (NWT). Such mining methodologies were already successfully evaluated against MT quality in [25]. The results of such mining are shown in Table 3.

Table 3: Number of obtained Bi-Sentences

Mining Method	Number of Bi-Sentences
MONO	21,132
BI	23,480
NW	24,731
NWT	27,723

As presented in Table 3, each of the improvements increased the number of parallel sentences discovered. In addition, in Table 4 a speed comparison is made using different versions of the tool.

Table 4: Computation Time of Different Yalign Version

Mining Method	Computation Time [s]
Y	92.37
MY	15.1
NWMY	18.2
GNWMY	16.4

A total of 1,000 comparable articles were randomly selected from Wikipedia and aligned using the native implementation (Y), multi-threaded implementation (MY), classifier with the Needleman-Wunch algorithm (NWMY), and with a GPU-accelerated Needleman-Wunch algorithm (GNWMY)

The results indicate that multi-threading significantly improved speed, which is very important for large-scale mining. As anticipated, the Needleman-Wunch algorithm decreases speed. However, GPU acceleration makes it possible to obtain performance almost as fast as that of the multi-threaded A* version. It must be noted that the mining time may significantly differ when the alignment matrix is big (text is long). The experiments were conducted on a hyper-threaded Intel Core i7 CPU and a GeForce GTX 660 GPU.

6. Evaluation of obtained comparable corpora

Using techniques described above, we were able to build comparable corpora and mine them for parallel sentences for the four languages being part of IWSLT 2015 evaluation campaign. We used GPU accelerated Needleman-Wunch algorithm, the classifier was tuned and Wikipedia page titles were downloaded separately. We focused on DE, FR, CS and VI. The corpora statistics are presented in Table 5.

Table 5: Results of mining after improvements

Language Pair	Number of bi-sentences	Number of unique EN tokens	Number of unique foreign tokens
DE-EN	2,459,662	2,576,938	2,864,554
FR-EN	818,300	1,290,000	1,120,166
CS-EN	27,723	98,786	104,596
VI-EN	58,166	92,434	93,187

To evaluate the corpora, we trained baseline systems using IWSLT 2015 official data sets and enriched them with obtained comparable corpora, both as parallel data and as language models. The enriched systems were trained with the baseline settings but additional data was adapted using linear interpolation and Modified Moore-Lewis [23]. Because of the well know MERT instability, tuning was not performed in the experiments [20]. Using MERT would most likely improve overall MT systems quality but it some cases it could produce false positive results, what needed to be avoided in order to properly evaluate only the impact of augmented corpora [20].

Table 6: Results of MT Experiments

LANGUAGE	SYSTEM	DIRECTION	BLEU
DE-EN	BASE	→EN	30.21
	EXT	→EN	31.37
	BASE	EN←	21.07
	EXT	EN←	22.47
FR-EN	BASE	→EN	35.95
	EXT	→EN	37.01
	BASE	EN←	35.73
	EXT	EN←	37.79
CS-EN	BASE	→EN	23.55
	EXT	→EN	24.09
	BASE	EN←	14.05
	EXT	EN←	14.93
VI-EN	BASE	→EN	22.84
	EXT	→EN	23.38
	BASE	EN←	26.23
	EXT	EN←	26.76

The evaluation was conducted using official test sets from IWSLT 2010-2013 campaigns and averaged. For scoring purposes, Bilingual Evaluation Understudy (BLEU) metric was used. The results of the experiments are shown in Table 6. BASE in the Table 6 stands for baseline system and EXT for enriched systems.

As anticipated, additional data sets improved overall translation quality for each language and in both translation directions. The gain in quality was observed mostly in the English to foreign language direction.

7. Conclusions

Bi-sentence extraction has become more and more popular in unsupervised learning for numerous specific tasks. This method overcomes disparities between English and other languages. It is a language-independent method that can easily be adjusted to a new environment, and it only requires parallel corpora for initial training. Our experiments show that the method performs well. The resulting corpora increased MT quality in a wide text domain. In some cases, only very small BLEU score differences were reported. Nonetheless, it can be assumed that even small differences can make a positive influence on real-life, rare translation scenarios. In addition, it was proven that mining data using two classifiers trained from a foreign to a native language and vice versa, can significantly improve data quantity, even though some repetitions are possible. From a practical point of view, the method requires neither expensive training nor language-specific grammatical resources, but it produces satisfying results. It is possible to replicate such mining for any language pair or text domain, or for any reasonable comparable input data.

8. 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 PJATK statutory resources ST/MUL/02/2015.

9. References

[1] WOŁK, K.; MARASEK, K. Real-Time Statistical Speech Translation. In: *New Perspectives in Information Systems and Technologies, Volume 1*. Springer International Publishing, 2014, p. 107-113.

[2] WOŁK, K.; MARASEK, K. Polish–English Speech Statistical Machine Translation Systems for the IWSLT 2013. In: *Proceedings of the 10th International Workshop on Spoken Language Translation, Heidelberg, Germany*. 2013, p. 113-119.

[3] KOEHN, P. *Statistical machine translation*. Cambridge University Press, 2009.

[4] BERROTARÁN G., CARRASCOSA R., VINE A., Yalign documentation, https://yalign.readthedocs.org - accessed 01/2015

[5] DIENY R., THEVENON J., MARTINEZ-DEL-RINCON J., NEBEL J.-C. Bioinformatics inspired algorithm for stereo correspondence. *International Conference on Computer Vision Theory and Applications*, March 5–7, Vilamoura - Algarve, Portugal, 2011.

[6] MUSSO, G. Sequence alignment (Needleman-Wunsch, Smith-Waterman), http://www.cs.utoronto.ca/~brudno/bcb410/lec2notes.pdf.

[7] ADAFRE, S.; DE RIJKE, M. Finding similar sentences across multiple languages in wikipedia. In: *Proceedings of the 11th Conference of the European Chapter of the*

Association for Computational Linguistics, 2006, p. 62-69.

[8] MOHAMMADI, M.; GHASEMAGHAEE, N. Building bilingual parallel corpora based on wikipedia. In: *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on*. IEEE, 2010, p. 264-268.

[9] TYERS, F. M.; PIENAAR, J. A. Extracting bilingual word pairs from Wikipedia, *Collaboration: interoperability between people in the creation of language resources for less-resourced languages 19*, 2008, p. 19-22.

[10] YASUDA, K.; SUMITA, E. Method for building sentence-aligned corpus from wikipedia. In: *2008 AAAI Workshop on Wikipedia and Artificial Intelligence (WikiAI08)*, 2008, p.263-268.

[11] SMITH, J. R.; QUIRK, C.; TOUTANOVA, K. Extracting parallel sentences from comparable corpora using document level alignment. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, p. 403-411.

[12] PAL, S.; PAKRAY, P.; NASKAR, S. K. Automatic Building and Using Parallel Resources for SMT from Comparable Corpora. In:*Proceedings of the 3rd Workshop on Hybrid Approaches to Translation (HyTra)@ EACL*, 2014. p. 48-57.

[13] PLAMADA, M.; VOLK, M. Mining for Domain-specific Parallel Text from Wikipedia. Proceedings of the Sixth Workshop on Building and Using Comparable Corpora, *ACL 2013*, 2013, p.112-120.

[14] AKER, A.; KANOULAS, E.; GAIZAUSKAS, R. J. A light way to collect comparable corpora from the Web. In: *LREC*, 2012, p. 15-20.

[15] STRÖTGEN, J.; GERTZ, M.; JUNGHANS, C.. An event-centric model for multilingual document similarity. In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, p. 953-962.

[16] PARAMITA, M. L., et al. Methods for collection and evaluation of comparable documents. In: *Building and Using Comparable Corpora*. Springer Berlin Heidelberg, 2013, p. 93-112.

[17] WU, D.; FUNG, P. Inversion transduction grammar constraints for mining parallel sentences from quasi-comparable corpora. In: *Natural Language Processing–IJCNLP 2005*. Springer Berlin Heidelberg, 2005, p. 257-268.

[18] CETTOLO, M.; GIRARDI, C.; FEDERICO, M. Wit3: Web inventory of transcribed and translated talks. In: *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*. 2012, p. 261-268.

[19] CLARK, J. H., et al. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, 2011, p. 176-181.

[20] JOACHIMS, T.. *Text categorization with support vector machines: Learning with many relevant features*. Lecture Notes in Computer Science vol 1398, 2005, p. 137-142.

[21] WOŁK, K.; MARASEK, K. A Sentence Meaning Based Alignment Method for Parallel Text Corpora Preparation. In: *New Perspectives in Information Systems and Technologies, Volume 1*. Springer International Publishing, 2014, p. 229-237.

[22] AXELROD, A.; HE, X.; GAO, J. 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, p. 355-362.

[23] ROESSLER, R. A GPU implementation of Needleman-Wunsch, specifically for use in the program pyronoise 2. *Computer Science & Engineering*, 2010.

[24] WOŁK, K.; MARASEK, K. Tuned and GPU-accelerated parallel data mining from comparable corpora. In: *Text, Speech, and Dialogue*. Springer International Publishing, 2015, p. 32-40.

[25] MOORE, Robert C.; LEWIS, William. Intelligent selection of language model training data. In: *Proceedings of the ACL 2010 conference short papers*. Association for Computational Linguistics, 2010, p. 220-224.

[26] KHALADKAR C. S. An Efficient Implementation of Needleman Wunsch Algorithm on Graphical Processing Units, PHD Thesis, School of Computer Science and Software Engineering, The University of Western Australia, 2009.

[27] https://github.com/machinalis/yalign/issues/3 accessed 10.11.2015

[28] HAGHIGHI, A., et al. Better word alignments with supervised ITG models. In:*Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, p. 923-931.

Applying Cross-Entropy Difference for Selecting Parallel Training Data from Publicly Available Sources for Conversational Machine Translation

William D. Lewis, Christian Federmann, Ying Xin

Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
{wilewis,chrife,v-yixi}@microsoft.com

Abstract

Cross Entropy Difference (CED) has proven to be a very effective method for selecting domain-specific data from large corpora of out-of-domain or general domain content. It is used in a number of different scenarios, and is particularly popular in bake-off competitions in which participants have a limited set of resources to draw from, and need to sub-sample the data in such a way as to ensure better results on domain-specific test sets. The underlying algorithm is handy since one can provide a set of in-domain data and, using a language model (LM) trained on this in-domain data, along with one trained on out-of-domain or general domain content, use it to "identify more of the same." Although CED was designed to select domain-specific data, in this work we are generous regarding the notion of "domain". Instead of looking for data of a particular domain, we seek to identify data of a particular *style*, specifically, data that is conversational. Our interest is to train conversational Machine Translation (MT) systems, and boost the available data using CED against large, publicly available general domain corpora. Experimental results on conversational test sets show that CED can greatly benefit machine translation system quality in conversational scenarios, and can be used to significantly increase the amount of parallel conversational data available.

1. Introduction

Cross Entropy Difference (CED) as defined by [1] has proven to be a very effective method for selecting domain-specific data from a larger corpus of out-of-domain or general domain content. It is used in a number of different scenarios, and is particularly popular in bake-off competitions—such as those hosted by the WMT [2] or IWSLT [3]—in which participants have a limited set of resources to draw from, and need to sub-sample the data in such a way as to ensure better results on domain-specific test sets. It has also proven useful in scenarios where training on all available data is not possible or feasible, or where iterating on large samples of data takes too long [4].

The algorithm is handy since one can provide a set of in-domain data and, using an LM built over the in-domain data, use it to "find more of the same" in a larger store of parallel or monolingual data. Although the output generated by CED may not truly be *in-domain*—Axelrod et al 2011 [5] use the term "pseudo in-domain"—the resulting data generally proves useful enough, and quality on relevant, in-domain, test data improves sufficiently enough, to warrant CED's inclusion in one's "bag of tricks" for manipulating data for SMT or language model building.

Although CED was designed to select domain-specific data, in this paper we are generous regarding the notion of "domain". Since we are looking for data not necessarily of a particular domain but rather we are looking for data of a particular *style* or *register*, that is, conversational. People have conversations about just about anything, so conversations truly defy domain.

Our primary interest, however, even more than using CED for style adaptation, is to find a means to bolster the amount parallel conversational data that is available for training conversational MT systems—essentially MT systems that we could be used in an end-to-end speech-to-speech (S2S) pipeline. Conversational data, specifically fluent transcripts of conversations, especially *parallel* conversational data, is very difficult to come by; only a very small set of language pairs have any parallel conversational data, and the quantities that are available are quite small. By contrast, the amount of broad-domain parallel data that is available has grown dramatically over the past few years (*e.g.,* CommonCrawl, EuroParl, United Nations, etc.). Enter CED as a method to find conversational content in the much larger stores of heterogeneous, general domain data.

We assume that a conversational MT system must be able to take as input the transcripts of speech recognition (*a la* [6]). We assume further that we have a mechanism to clean up disfluencies in the source ASR output in order to make it more hospitable to an MT engine (how to do such data cleaning is beyond the scope of this paper;[1] we assume clean input

[1] We employ a method for such data cleaning called *TrueText*. [7] gives some background on how producing "fluent" content from speech recognition can improve downstream processes, such as Machine Translation. Given space limits, we will not expand upon TrueText in this paper, but suggest the reader explore [7] for more background.

for the MT, effectively constituting "oracle" output from the ASR[2]. To this end, we seek to use CED to bolster the amount of *parallel* conversational-style (or "pseudo-conversational-style") data available to us. Using a method to discover conversational content, notably *parallel* conversational content, can help build more robust conversational MT systems.

To determine the utility of data output by CED for this task, we measure end-to-end MT results on conversational test sets representative of actual mono- and bi-lingual conversations. For our general domain corpora, we draw from all publicly available parallel sources for English↔French that we know of (shown in detail in Table 3). Combined and added to training, these sources act as our general domain source data and our ceiling (when we train on all of the data). To test a "what if" scenario—that is, "what if" we had a much larger store of data available to draw from beyond those that are publicly available—we use CED against a very large store of Web-scraped English↔French content (over 500 million parallel sentences) combined with the publicly available data to create another ceiling. With this ceiling we show that CED can expand to much larger stores of data, and demonstrate the gain others can reasonably expect to see using this method in the near-term. Experimental results on conversational test sets show that style adaptation using CED greatly benefits MT quality in conversational scenarios.

This paper is organized as follows: Section 2 provides more details on the CED method while Section 3 explains our experimental setup and the data we have used. We discuss results in Section 4 and conclude with a summary and an outlook to future research questions in Section 5.

2. Background

2.1. Cross-Entropy Difference

The intent of the Cross Entropy Difference (CED) algorithm [1] is to identify a subset of data in a much larger corpus of data that is in the "domain" of interest. Using an in-domain corpus, and an LM built over the corpus, we can find more content that resembles the domain of interest. The CED algorithm, as shown in Figure 1, relies on three principal components: (i) an in-domain LM S_{in} (or LMs, in the case of [5]), (ii) an out-of-domain LM S_{out}, and (iii) an out-of-domain or general domain corpus from which we are selecting data ((ii) can be built over the data in (iii), but that is not required). For each sentence in (iii) s_i, the CED algorithm calculates the cross entropy from the in-domain LM S_{in}, and subtracts from it the cross entropy for the same sentence scored against the out-of-domain LM S_{out}. Although one would expect scoring against the in-domain should be adequate in and of itself, *e.g.,* one would expect the entropy of sentences that share characteristics of the domain, *e.g.,* shared n-gram frequencies, would be adequately scored against the in-domain LM S_{in}. This is the thinking behind related and earlier attempts at the same [8, 9]. However, by simultaneously scoring against an LM built over content that is not in the domain of interest, we favor content that scores *better* on the in-domain LM and more *poorly* against the out-of-domain LM. This, in effect, "pushes" the selection towards in-domain content and away from out-of-domain content. Figure 1 shows the algorithm.

$$\mathcal{CED}(s_i|S_{\text{in}}, S_{\text{out}}) = H_{\mathcal{LM}(S_{\text{in}})}(s_i) - H_{\mathcal{LM}(S_{\text{out}})}(s_i) \quad (1)$$

The most common usage of CED in MT, as noted earlier regarding *bake-offs*, has been to find additional content in a particular domain, say "news text", in an out-of-domain corpus, say "parliamentary proceedings", *e.g.,* Europarl [10]. We may or may not have bilingual data for the in-domain corpus, but if we do we can pool it with a set of data selected by CED, and use it for training our in-domain translation models. The percentage of content that we should select is often decided upon by trial and error, that is, select 5%, 10%, 15%, etc., of the data desired, and where quality plateaus, select that percentage. Since CED assigns a score to every sentence for an out-of-domain corpus, we can rank the data by that score, and select the top n% from the ranked data, and then train our models on that percentage.

2.2. The Nature of Conversational Data

The definition of what constitutes a domain has mostly been avoided in the MT literature. Researchers will generally refer to a domain by name, *e.g.,* news, blogs, government, tech, etc., without ever really defining what the characteristics of that domain are. For conversational data, which is really not a domain at all but rather a *style* or *register*, *i.e., a manner in which language is used*, we can be a little clearer in our definition. There are a number of features that characterize data in the conversational style, among them being what is shown in Table 1. Given that most of these features can be captured by simple LMs, their presence can be boosted by CED.

3. Data and Experiments

3.1. Data Sources (for Training and Tuning)

In this section we provide detail on the data we use in our experiments:

Publicly available data sets – Table 2 shows the sets of data that are available publicly as well as their sizes. This data serves as our general-domain content (our S_{out}) for the set of experiments against which we apply CED (and we also use it for producing our Ceiling System (D), and we randomly sample it for control baselines (B)).

CED seed data – Our seed, in-domain corpus is drawn from the Fisher Corpus [11], and consists of 760K English sentences. The Fisher Corpus consists of transcripts

[2]See [6] for an alternative approach.

Id	Feature	Description / Examples
F1	Increased use of contracted forms	*don't, can't, I'm, I'll, you're*
F2	Increased use of reduced forms	Forms common in colloquial speech, *e.g., gonna, wanna, shoulda, musta, kinda*
F3	Increased use of slang	
F4	Higher frequency of 1st and 2nd person	1st and 2nd person pronouns and verbal forms are more common in colloquial speech vs. Web content as a whole
F5	Shorter Sentences	Conversational utterances tend to be shorter than many sources of textual content
F6	Reduced vocabulary	
F7	Sentence Fragments/Partial Utterances	
F8	Disfluencies and Restarts	Disfluencies: *um, uh, you know, I mean* Restarts: *I I, I'm uh I've*

Table 1: Features of the conversational style

Source	Sentences	Words (English)
Common Crawl 2015	2.98M	58M
Europarl v7 2015	1.79M	43M
FBIS	38K	851K
Gutenberg (No Shakespeare)	196K	3.1M
JRCDGT	698K	15.8M
JRC	1.87M	45.3M
MultiUN	9.1M	228.6M
Subtitle2012	13.8M	96.8M
Subtitle2013	15.1M	106.6M
WIT3	167K	2.5M
WMT2009 Giga	23.93M	532.8M
WMT2009 News	64.6M	1.33M
WMT2011 News	117K	2.5M
WMT2012 News	139K	2.91M
WMT2013 News Commentary	158K	3.4M
WMT2014 News Commentary 2015	179K	3.8M
Total	70.4M	1.15B

Table 2: Publicly available data comprising our general pool

of over 2,000 hours of English-speaking phone calls. These are unscripted and, hence, very conversational.

Training data – Core to one of our baseline systems (A) is just the set of Open Subtitle content. We assume that subtitle data is reasonably conversational (albeit scripted), and thus makes a good "core" set of training data for conversational MT. It acts as our primary baseline. To (A), we add varying amounts of "Style Adapted" (SA) data. Our SA data consists of four different sets, specifically 10%, 20%, 30%, and 40%, ranked by CED, drawn from the publicly available data shown in Table 2.[3] Our Random Baseline system (B) consists of a random sample of our public data, with approximately the same word count as (A). To be a useful control against (A), we again add the 10-40% SA samples. Baseline (C) is a system containing just the 20% SA sample, and nothing else. Its word count is approximately the same as (A) and (B), and thus can be used for comparison purposes. System (D) was trained on *all* publicly available training data, and thus should act as a *ceiling* system, *possibly* reflecting the peak BLEU scores we might expect to achieve. Finally, system (E) is a system consisting of a very large SA sample, paired with OpenSubtitle content at its core (same assumption as (A) as to the underlying value of subtitle content for conversational systems). The data consists of approximately 94M parallel sentences. The SA data for (E) was drawn from a very large corpus of English↔French Web content, plus all publicly available sources, clocking in at greater than 500 million parallel sentences. We were unable to train another ceiling system on all of this data, so the style-adapted system (E) effectively acts as another ceiling system. The sentence and word counts for each baseline system (A), (B), and (C) are shown in Table 3. We also include the sizes of our two ceiling systems, (D) and (E).

[3] In production, we select the sample, *e.g.,* , 10%, 20%, etc., that produces the highest BLEU score for the particular task at hand. See [5] or [12] for further exploration of the methodology.

Data set	Sentences	Words
Baseline (A)	22,912,400	167,690,601
Baseline (B)	7,288,000	167,127,882
Baseline (C)	14,300,000	166,085,537
Ceiling (D)	60,864,815	1,037,969,219
Ceiling (E)	93,700,367	1,145,178,939

Table 3: Overview on training data sets for our experiments

Tuning data – Our dev set is based on a random sample of Web content which contains 6,870 sentence pairs and a total of 123,030 English and 132,903 French words, respectively. Based on our experience with this data set, it can be considered *lightly conversational* as it shares some of the characteristics of conversational

data. Still, our main tuning target is general domain text so any measurable improvements on our *strictly conversational test sets* will effectively prove that our data selection approach works as desired.

3.2. Test Data

To test the impact of our data selection on resulting SMT systems, we built several test sets. These are listed below. Crucially, since we wanted to measure the performance of SMT systems on true, open-domain conversational content, each of the **Speech** test sets was created from actual Skype calls that were recorded between participants who were either speaking the same language or different languages (in the latter case, drawn from bilingual conversations).

Supporting real-time, open-domain, bilingual conversations is the gold standard for S2S systems. To evaluate a conversational MT system that performs the translation function in such a system, we felt our test sets had match the scenario as close as possible, that is, be representative of open-domain, conversations. To that end, $\textbf{Speech}_{EX,1}$ and $\textbf{Speech}_{XE}$ consist of transcripts of English→French and French→English bilingual conversations, respectively, which were then translated into the opposing languages. These tests sets are relatively hard, since they consiste of true, real-time bilingual conversations, but they are also representative of our ultimate S2S goal: to support free-form, open-domain, bilingual conversations between *monolingual* speakers.

1. $\textbf{Speech}_{EX,1}$ – This test set consists of the transcripts of the English side of bilingual English↔French conversations. Participants were English↔French bilinguals, who were fully conversant in both languages. In each conversation, one of the two consistently spoke English, the other spoke French. The English transcripts were normalized and then translated into French.
2. $\textbf{Speech}_{EX,2}$ – This test set consists of the transcripts of the English side of bilingual English↔French conversations conducted by monolingual speakers, mediated by an S2S system, namely Skype Translator.[4] In other words, each participant spoke in their own language, and the S2S system transcribed and translated their spoken content into the other language. The English audio was human transcribed (the test data does not contain ASR output), normalized, and then translated into French.[5]
3. $\textbf{Speech}_{XE}$ – This test set consists of the French side of bilingual English↔French conversations. It is effectively the equivalent of $\text{Speech}_{EX,1}$, except in this case the French side data was kept and translated into English. All French data has been recorded by French native speakers so it is an accurate representation of conversational French.
4. $\textbf{Eval2000}_{EX}$ – Eval2000 [13] is a standard speech test set consisting of transcripts of English phone conversations. We translated a sample of the Eval2000 transcripts into French in order to create this test set.
5. $\textbf{Social}_{XE}$ – This test set consists of a sample of French Facebook posts, which were then translated into English. Although not strictly conversational, Facebook posts, as with any other social media, exhibit some of the features one sees in conversational transcripts.
6. **WMT2013** – This test set consists of a sample of standard test set used at the 2013 Workshop on Machine Translation [2]. It acts as a sanity check. It contains content that is really not relevant to the conversational MT style.

3.3. Experimental Setup

In order to measure the effectiveness on translation quality of data selected using CED, we ran a series of experiments drawing from a general domain pool of English↔French data (our S_{out}). All of the data is publicly available, consisting of corpora such as the CommonCrawl, Project Gutenberg, various WMT data sets, UN data, etc., which are broken down in Table 2.[6] In total, this corpus consists of approximately 70M sentence pairs and 1.15B words (English side), before removing duplicates. The in-domain (or "in-style") data, or seed data (S_{in}), which is constant in these experiments, consists of a 760K sentence sample from the Fisher data set [11]. Fisher consists of transcripts of *unscripted* phone calls, so the data are quite conversational, and very similar to the **Speech** test sets. We also include in our experiments two ceiling systems, trained on the following data: The first is trained on all available publicly available corpora (effectively, all sources shown in Table 2). The second is a *"what if"* system, trained on 94M sentences, including some 24M sentences discovered using CED from a very large scrape of the Web, consisting of over 500M sentence pairs, which is then combined with other conversational content. The intent of the second ceiling is to demonstrate the potential of CED on very large corpora, and to provide a proof of concept of what is possible as more data becomes publicly available (*e.g.,* as the CommonCrawl data continues to grow). The hypothesis is that as more data becomes available, there will be more snippets of conversational data in the general pool, which increases the amount of beneficial data we extract when we run CED. This in turn will benefit those who are building conversational S2S and MT systems.[7]

[4] Skype Translator is available at the following URL: http://www.skype.com/en/translator-preview/. The functionality of Skype Translator is also being integrated into other Skype versions.

[5] We assume $\textbf{Speech}_{EX,2}$ is easier than $\textbf{Speech}_{EX,1}$, since users were bound by the current state of the art of the S2S at the time the recordings were made.

[6] Much of this data, specifically, the Europarl data, the CommonCrawl parallel data, and any data sets labeled with "WMT" are available from WMT 2015 [14] at `http://statmt.org/wmt15/translation-task.html`. WIT3 comes from IWSLT [15].

[7] Crucially, CED can be run on corpora of any size. Realistically, the only limiting factors are disk space, the amount of time to run the algorithm over

Our basic experimental setup compares a baseline MT system trained on subtitle data (A) to a contrastive system trained on a set of randomly selected general domain data, basically parallel text harvested from the Web, of approximately the same word count (B). We assume (A) to be conversational (albeit, scripted conversations). To each baseline, we incrementally add samples of style-adapted data, generated using CED from S_{out}. We have an additional baseline of just style-adapted data of similar sizes to (A) and (B), which is composed of just style-adapted content (C). (C) provides a baseline that demonstrates what is possible in conversational MT just using CED (and is size-controlled, having roughly the same sentence count as (A) and (B), and thus directly comparable to these systems). Finally, we train a system on all available general domain data, to act as a "ceiling" (D). All systems are compared against multiple conversationally oriented test data, with a sanity check test set from the WMT, specifically a test set sampled from the WMT 2013 English↔French test data [2].

We use custom tree-to-string (T2S) systems for training the models for our engines. We require a source-side parser for our T2S decoder, which we have for both English and French; for the English→French direction, we use the English parser, and for the opposing direction, the French one. 5-gram Language Models (LMs) are trained over the target-side data for each system. We use Minimum Error Rate Training (MERT) [17] for tuning the lambda values for all systems, and we report results in terms of BLEU score [18] on lowercased output with tokenized punctuation.

4. Evaluation and Analysis

4.1. Experimental Results

Looking at Tables 4 and 5, it is fairly clear that trainings performed on conversational training data fare well on test sets that are conversational in nature. This should not come as a surprise. However, there are some surprises in the results. For the English→French trainings, baseline (C) which consists of just the style-adapted data, outperformed *all* other trainings on the EX Speech-related test sets (having scores of 52.39, 47.39, and 35.45 for $Speech_{EX,1}$, $Speech_{EX,2}$, and *Eval2000*, respectively), even besting systems trained with subtitle data, including those trained with additional CED "style-adapted" (SA) data (best in class for each test EX set: 51.68 with 20% SA data, 46.59 with 40% SA data, and 34.31 with 10% SA data). What was most startling, however, was that the Random baseline (B) bested the subtitle Baseline (A) on all EX test sets, scoring 50.79, 45.09 and 35.23 versus 50.28, 44.63 and 32.77. This suggests that the subtitle data, contrary to our initial assumptions, is not a good baseline for a conversational MT system. Further, adding SA data for the Random baselines did sometimes improve scores on EX test sets, but no Random baseline+SA pairing bested Baseline (C)—that is, SA data *alone* beats any random baseline—on EX test sets.

Results for the English→French trainings on the XE test sets paint a different story, however. As noted in Section 4.2, there are two XE test sets, $Speech_{XE}$ and $Social_{XE}$. The former consists of the French side of English→French conversations, and the latter consists of French Facebook posts. Both sets of data were translated into English. On the $Speech_{XE}$ test set, the subtitle Baseline (A) beats *all* other results (excepting Big Data (E)), including SA (D) and any combination of SA with Subtitle (A) or Random (B); the baseline score of 51.61 is beat by no training other the Big Data (E). So, contrary to the assessment that subtitle data makes a poor baseline system, it actually proves to be very good *when the data is French sourced*. In fact, it proves to be a much better baseline than SA data (Baseline (C)), completely the opposite of what we saw on the EX test sets. (We examine what the source of this "directionality bias" might be in more detail in Section 4.3.2). On the $Social_{XE}$ test set, the SA baseline (C) does equally poorly, beating only the random baseline (23.45 vs. 22.76). Again, since the $Social_{XE}$ is French sourced, it provides further evidence of some sort of directionality bias.

For the French→English trainings, the subtitle baseline trainings (A) fare much better than the equivalent EX trainings: on all conversational test sets, they best the SA baseline (C), in some cases paired with varying quantities of SA data. The only odd result is the performance of the Random baseline (B) when paired with 30% SA for $Speech_{EX,2}$, which does the best of any system outside of (E). Baseline (B) does very poorly by itself on all test sets, however, performing better when paired with the SA data. SA data, thus, proves to be a useful augmentation for the random baseline (B). SA proves less useful for the subtitle baseline (A) on the EX speech test sets, but much better for the XE speech test set (and the social media XE test set as well). Again, there is evidence here for some sort of directionality bias.

Overall, the SA data contributes. By itself, in the EX trainings, it has proven essential. For XE, it's a useful addition to subtitle data when measured against XE test sets.

4.2. Overview of Experimental Results

Subtitle data appears less useful, but only when either (a) English sourced data is used or (b) training English→French systems. In all other cases, subtitle data proves useful for training conversational MT systems. Domain adapted data, however, proves highly useful for training conversational MT systems. Using existing and readily available public sources of English↔French data, and using existing and readily available monolingual, conversational English seed data, we are effectively able to select "conversational" data from these sources in order to train conversational MT systems with higher BLEU scores. Although SA data has proven universally useful, its value differs depending on the direction of training or test data. In the next section, we examine some

the data—LM scores do not have to be stored in memory, but can be output directly—and building the out-of-domain or general domain LM. Using KenLM [16] for the latter makes CED feasible in most scenarios.

English→French							
Experiment		Test sets					
Data	System	Speech$_{EX,1}$	Speech$_{EX,2}$	Speech$_{XE}$	Eval2000$_{EX}$	Social$_{XE}$	WMT13
Baseline (A)	OpenSubtitle	50.28	44.63	**51.61**	32.77	25.27	28.87
+10% SA	OpenSubtitle	51.37	45.36	51.59	*35.02*	25.46	30.80
+20% SA	OpenSubtitle	*51.68*	46.46	51.39	33.95	25.87	**31.43**
+30% SA	OpenSubtitle	51.49	46.54	51.35	33.60	25.75	31.38
+40% SA	OpenSubtitle	51.35	*46.59*	51.07	33.74	**26.10**	31.39
Baseline (B)	Random	50.79	45.09	46.59	*35.23*	22.76	30.39
+10% SA	Random	51.23	46.13	49.68	33.22	24.18	30.50
+20% SA	Random	50.90	45.51	51.07	34.18	**26.10**	30.86
+30% SA	Random	*51.74*	*46.53*	*51.18*	33.87	25.33	*30.97*
+40% SA	Random	51.19	46.19	50.72	33.38	25.40	30.96
Baseline (C)	SA Only	**52.39**	**47.39**	46.45	**35.45**	23.45	30.40
Ceiling (D)	All	50.55	45.86	50.47	32.98	25.67	31.22
Big Data (E)	S2S	58.32	54.04	52.87	37.23	26.65	32.72

Table 4: Translation quality measured using BLEU scores for language pair English→French. Best scores per experiment *in italics*, globally best scores **in bold face**. Table compares Baseline system trained on General domain data to *pseudo in-domain* DomainAdapt system trained on data obtained using the CED method.

distributional clues as to why SA data is useful, and what may be causing this directional discrepancy. The next section constitutes a very preliminary analysis of some of the data and some of the features. We intend to expand this work in the future. What is clear, however, is that there is some sort of directionality bias, and that this bias interacts with the sources of the data.

4.3. A Quick Look at the Conversational Style Features in CED Output

In this section, we look at two main issues: First of all, we look at the distribution of some of the values for a subset of the conversational features, as described in Table 1, across our subtitle, style-adapted, and random baselines, as well as the Fisher corpus we used as our seed data. Second, we compare the distribution of examples of these features in French as well, to see if there are potential discrepancies. We then propose a hypothesis of what might be causing the directionality bias.

4.3.1. Distribution of Conversational Features

In Table 6 we look at the distribution of a subset of the features described in Table 1, specifically, Contractions (F1), Reductions (F2), and 1st and 2nd person forms (F4) (these too are contractions, thus overlap with F1). A comparison between the Subtitle, SA 20%, and the General (Random Sample) shows some interesting tendencies. All three are controlled such that their word counts are roughly the same; the counts in Table 6 are thus effectively normalized (the Fisher data stands out in this regard since it is smaller, and thus is effectively not normalized). Contracted forms, Reductions, and the Distribution of 1st and 2nd person forms are much more frequent in the Subtitle data, suggesting that, if these values are true indicators of conversational content, it is far more conversational. The SA 20% data set is not quite as strong as Subtitle in these feature sets, but it is much stronger than the General data set in both Contracted and 1st and 2nd person forms. Since both SA 20% and the General data were sampled from the same General pool, this provides strong evidence that the CED algorithm, drawing from distributional clues in the Fisher seed data, is selecting a better sample of data for the conversational setting than a random sample does.[8] Noticeably weak in the SA 20% sample are reduced forms, suggesting that they do not occur frequently in the general domain pool (and thus are not available for CED to discover). Thus, in summary, as long as we accept that the distribution of feature values listed here are representative of conversational content, subtitle data does appear to be highly conversational, in comparison with the other data, with the SA 20% data coming in second. These data, in and of themselves, however, do not explain the directionality bias.

4.3.2. The Directionality Bias

We observed in Section 4.1 that our English→French baseline (A) trainings do poorly on English-sourced test data as

[8] It would appear that the LM is, in fact, boosting conversational content based on scoring against the Fisher LM, boosted further by CED due to the absence of these values in the general pool (since those scores are subtracted from the former by CED).

French→English							
Experiment		Test sets					
Data	System	Speech$_{EX,1}$	Speech$_{EX,2}$	Speech$_{XE}$	Eval2000$_{EX}$	Social$_{XE}$	WMT13
Baseline (A)	OpenSubtitle	**55.04**	48.49	51.84	**36.57**	26.77	29.43
+10% SA	OpenSubtitle	54.70	*48.70*	52.60	36.34	27.24	31.61
+20% SA	OpenSubtitle	54.64	48.30	**53.54**	36.22	*27.43*	31.97
+30% SA	OpenSubtitle	53.56	47.61	52.93	35.62	26.89	32.32
+40% SA	OpenSubtitle	53.29	47.67	52.56	35.61	27.36	*32.45*
Baseline (B)	Random	48.33	43.12	46.52	31.81	23.63	31.39
+10% SA	Random	54.11	48.28	*52.78*	35.32	26.59	32.01
+20% SA	Random	53.36	48.39	52.70	35.58	27.07	32.06
+30% SA	Random	*54.39*	**48.79**	52.54	*35.91*	*27.39*	32.27
+40% SA	Random	54.06	48.19	52.64	35.40	27.25	*32.31*
Baseline (C)	SA Only	49.44	44.05	49.37	32.35	23.85	31.48
Ceiling (D)	All	53.73	47.40	52.88	35.38	**27.73**	**32.54**
Big Data (E)	S2S	57.80	51.71	55.54	37.25	27.32	33.30

Table 5: Translation quality measured using BLEU scores for language pair French→English.

compared to our SA baseline (C), but trump baseline (C) for test sets that are French-sourced. Further, we observed that baseline (A) does well on all conversational test sets irrespective of sourcing for the French→English trainings; the baseline (A) trainings beat the SA baseline (C) in all cases. Only on the French-sourced Facebook test set, *Social*$_{XE}$, does baseline (C) show weaker results.

These puzzling results *could* be caused by the discrepancy in conversational features between the English and French sides of our training data. Although we will not find analogous contracted forms in the French, *e.g.,* , for the same person, verbal forms, etc., we can look at the distribution of values for similar features between the two languages. In Table 7 we show values for a small set of French features, namely, (F1) Contractions and (F3) Slang, and a small set of values for each. The (F1) feature is comparable to the same in English in Table 6; (F3) was not tabulated for English, but since the French *argot* forms are often reductions, they are somewhat comparable to (F2) Reductions. When we compare the two tables, Table 7 and Table 6, we can see a much clearer difference between the conversational data (whether seed, subtitle, or SA) and the general data: the ratio of conversational features between conversational vs. general is much larger in French than in English. There are at least two possible reasons for this: (1) English speech is far more colloquial than French, indicated by a higher number of colloquial expressions that occur in conversational data than in written content. Or (2), transcribed English is more likely to preserve the colloquialisms than is transcribed French. (2) could result either from difference in transcription rules between the two languages, or an unconscious bias by French transcribers to avoid transcribing colloquialisms, at least, to avoid transcribing them literally or phonetically.

How might that affect BLEU scores and contribute to a directionality bias? If the English side has a larger number of colloquial expressions, there may likewise be a larger ratio of many-to-one mappings between English and French than in the other direction. In other words, for any given French expression, there will be a higher likelihood of at least two mappings on the English side for that expression (with all the English expressions essentially meaning the same thing, just written differently). Take, for example, the English future marker *gonna*. In formal English, *gonna* is always written as *going to*. A speaker, referring to himself, might say *I'm gonna*, but would never write it that way—*I'm going to* would be the way to write it formally. However, a transcriber, wishing to be true to the input, especially, it would appear, when tasked with captioning movie content, is more likely to write *I'm gonna*. The most common French expression for either is *je vais*, which is the standard form; there is no formal/informal dichotomy for this term in French. In the English→French trainings, both *I'm gonna* and *I'm going to* would resolve to *je vais*, effectively creating a 2:1 mapping, which would have little or no consequence in evaluations on *conversational* test data for the English→French direction. However, in the reverse direction, the 1:2 mapping could lead to occurrences of both forms in the output, causing a failure to match against the test data in a certain percentage of cases, effectively causing a reduction in BLEU scores. Multiplying this effect across the multitude of conversational forms showing in English, and absent in French, could explain the discrepancies observed in the two different directions of the trainings against the test data.

Feature	Seed (Fisher)	Subtitle (A)	SA 20 (C)	General (B)
F1 – Contractions				
don't	81,997	412,479	31,797	9,846
can't	13,717	135,393	9,401	2,707
shouldn't	1,345	13,009	1,156	325
wouldn't	6,439	36,347	2,448	586
couldn't	3,616	26,697	2,767	713
they'll	2,925	2	1,221	295
he'll	1,529	11	535	144
she'll	591	3	161	57
they're	30,713	71,153	7,716	1,780
she's	6,778	77,729	1,452	395
he's	18,842	235,203	4,622	1,352
F2 – Reductions				
gonna	9,588	3,473	4	5
wanna	3,819	960	21	30
shoulda	1	27	2	1
coulda	29	36	2	5
woulda	1	35	1	1
musta	33	1,404	352	707
kinda	7,575	3,671	149	50
F4 – First/Second				
I'm	67,814	460,910	17,981	4,542
I'll	5,735	107	3,775	894
you're	23,699	288,031	13,722	3,508
you'll	1,375	80	7,626	2,232
we're	14,028	125,116	12,589	3,491
we'll	1,817	10	4,000	1,124

Table 6: Distribution of conversational features across different data sets (English-only)

5. Conclusion and Future Work

Overall, the CED algorithm performs well in selecting conversational data from a general pool, as evidenced by the results in both Tables 4 and 5. The algorithm appears to select data in the conversational style, preserving many of the features observed in the conversational source data in the sampled output. The distribution of conversational features in "style" adapted data is not as strong as for conversational data, such as subtitle data, but it still captures a larger sample of conversational features than an equivalently sized random sample does. As shown in the experimentation, "style-adapted" data, that is, data selected by CED, is conversational enough to boost the quality of conversational MT systems. Further, we show that given much larger stores of data, we see even more marked improvements. The continued expansion of the CommonCrawl parallel data, as well as other publicly available sources, can only benefit the larger S2S community as it will consequently increase the pool of readily available (pseudo-)conversational content.

Although we touched upon the directionality bias observed between the English→French vs. the French→English trainings, and hypothesized a potential transcription "bias" between the two languages, the evidence presented was not particularly strong. Since further experimentation with a much larger general pool of data, upwards of 500 million sentence pairs, is showing the same directionality bias effects[9], further investigation in reasons behind this bias is warranted. In our future work, we plan to continue investigating the bias, which includes the exploration of conversational style adaptation for additional languages. We also plan to look at a much more complete set of conversational features (as discussed in [7]). We are also now experimenting with applying CED using other seed sources of data, including data sampled from conversations of Skype Translator users.

References

[1] R. C. Moore and W. D. Lewis, "Intelligent Selection of Language Model Training Data," in *Proceedings of the ACL 2010 Conference Short Papers*, Uppsala, Sweden, July 2010. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=138756

[9] These experiments were not included in this paper.

Feature	Subtitle (A)	SA 20 (C)	General (B)
F1 – Contractions			
J'sais pas	171	4	1
T'sais	71	2	1
T'es	52,493	173	76
J'suis	1,071	17	6
M'en fiche	1,185	8	4
F3 – Slang (Argot)			
putain	22,307	72	31
merde	25,486	219	96
ma pote	19	2	1
mon pote	5,234	40	8
meuf	807	13	18

Table 7: Distribution of conversational features across different data sets (French-only)

[2] O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2013 Workshop on Statistical Machine Translation," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 1–44. [Online]. Available: http://www.aclweb.org/anthology/W13-2201

[3] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 11th IWSLT Evaluation Campaign, IWSLT 2014," in *Proceedings of the eleventh International Workshop on Spoken Language Translation (IWSLT), Lake Tahoe, CA*, Lake Tahoe, CA, December 2014, pp. 2–17. [Online]. Available: http://www.mt-archive.info/10/IWSLT-2014-Cettolo.pdf

[4] W. D. Lewis and S. Eetemadi, "Dramatically Reducing Training Data Size through Vocabulary Saturation," in *Proceedings of the Eighth WMT*, Sofia, Bulgaria, August 2013.

[5] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of EMNLP*, 2011, pp. 355–362. [Online]. Available: http://research.microsoft.com/en-us/um/people/jfgao/paper/2011-emnlp-camera-select-train-data.pdf

[6] G. Kumar, M. Post, D. Povey, and S. Khudanpur, "Some insights from translating conversational telephone speech," in *Proceedings of ICASSP*, Florence, Italy, May 2014. [Online]. Available: http://cs.jhu.edu/~gkumar/papers/kumar2014some.pdf

[7] E. Fitzgerald, *Reconstructing Spontaneous Speech*. Baltimore, Maryland: The Johns Hopkins University, 2009.

[8] J. Gao, J. Goodman, M. Li, and K.-F. Lee, "Toward a unified approach to statistical language modeling for chinese," in *ACM Transactions on Asian Language Information Processing*, 2002, pp. 3–33.

[9] D. Klakow, "Selecting articles from the language model training corpus," in *ICASSP 2000*, Istanbul, Turkey, 2000, pp. 1695–1698.

[10] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," in *MT Summit X: Proceedings of the Tenth Machine Translation Summit*, ser. MT Summit '05. Phuket, Thailand: Asia-Pacific Association for Machine Translation, 2005, pp. 79–86. [Online]. Available: http://mt-archive.info/MTS-2005-Koehn.pdf

[11] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a resource for the next generations of speech-to-text," in *Proceedings of 4th International Conference on Language Resources and Evaluation, LREC*, 2004, pp. 69–71. [Online]. Available: https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/lrec2004-fisher-corpus.pdf

[12] A. Axelrod, Q. Li, and W. Lewis, "Applications of Data Selection via Cross-Entropy Difference for Real-World Statistical Machine Translation,," in *Proceedings of the IWSLT 2012*, Hong Kong, China, December 2012.

[13] M. Przybocki and A. Martin, "2000 NIST Speaker Recognition Evaluation LDC2001S97," Web Download. Philadelphia: Linguistic Data Consortium, 2001. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2001S97

[14] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, M. Huck, C. Hokamp, P. Koehn, V. Logacheva, C. Monz, M. Negri, M. Post, C. Scarton, L. Specia, and M. Turchi, "Findings of the 2015 workshop on statistical machine translation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 1–46. [Online]. Available: http://aclweb.org/anthology/W15-3001

[15] M. Cettolo, C. Girardi, and M. Federico, "WIT3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of EAMT*, Trento, Italy, 2012, pp. 261–268. [Online]. Available: http://hltshare.fbk.eu/EAMT2012/html/Papers/59.pdf

[16] 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

[17] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st ACL*, Sapporo, Japan, 2003.

[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 ACL*, Philadelphia, PA, 2002.

Source Discriminative Word Lexicon for Translation Disambiguation

Teresa Herrmann, Jan Niehues, Alex Waibel

Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology, Germany
firstname.lastname@kit.edu

Abstract

This paper presents a source discriminative word lexicon that performs translation disambiguation for individual source words using structural features, such as context and dependency relations in the sentence. The individual translation predictions are combined into sentence scores that are used in N-best list re-ranking to improve the translation output of a state of the art phrase-based machine translation system. The approach is used to improve explicitly the translation of word categories that require grammatical agreement to hold in the target language after translation, e.g. pronouns, as well as subjects and verbs. The results show that the translation predictions provided by the source discriminative word lexicon increase the prediction accuracy by up to 10%. The translation quality can be improved by up to 0.6 BLEU points on English-German translation.

1. Introduction

Ambiguity of words is a big challenge for all natural language processing tasks. Already within the same language, words can be ambiguous with regard to their part-of-speech (***can**, n. - **can**, v.*), word sense (***bank**, n., financial institution - **bank**, n., side of a river*) or what they are referring to in the given context (*The monkey eats the banana. **It** is brown.*). For translation, such ambiguities pose an additional difficulty. Unless the very same ambiguity exists in the target language, the ambiguity needs to be resolved in order to generate the correct translation. When translating into German, for example, depending on the correct part-of-speech, word sense and antecedent in the sentence, the translation for each of those examples is a different one.

The word(s) indicating which is the correct word sense or antecedent for an ambiguous word in a given context, could occur in a more distant part of the sentence. That means long-range dependencies need to be considered in order to generate the correct translation. We propose a discriminative framework for modeling these dependencies utilizing any conceivable set of features for predicting the correct translation. We show the potential of this approach in detail on the third type of ambiguity mentioned above: The translation of pronouns, which is conditioned on the translation of the antecedent they refer to, since the pronoun in the target language needs to share the morphological properties of the antecedent in the target language.

An approach to explicitly performing anaphora resolution to uncover the pronoun-antecedent relationship for pronoun translation disambiguation was carried out in [1]. Their experiments motivated the present work, however the approach was adapted in the following ways: Instead of focusing only on third person pronouns, we include all personal pronouns and also take translations into other word categories into account. In order to allow for a more comprehensive exploration of the source discriminative word lexicon approach we apply it for translation disambiguation for all words and perform separate evaluation of the performance on pronouns. We further evaluate it on another difficult agreement task, the agreement of subject and verb in a sentence.

State-of-the-art machine translation systems struggle with these particular kinds of linguistic requirements [2]. Hence, we believe our approach can provide a comprehensive solution for many of these challenges where long-range dependencies have to be met in order to ensure congruency of linguistic features.

2. Related Work

Already one of the early statistical approaches performs word sense disambiguation by defining senses according to the different translations of a word [3]. Since then, several approaches integrate word sense disambiguation into phrase-based [4] or hierarchical [5] translation systems or use it in N-best list re-ranking [6]. Context features as well as dependencies have been used to perform word sense disambiguation for different close and distant language pairs [7, 8].

Apart from applying actual word sense disambiguation in machine translation, linguistic information, such as context words, dependencies or syntax can be integrated in machine translation as additional features in order to improve the translation quality [9, 10].

Among the approaches that particularly model translation prediction as is done in this paper, [11] predict the occurrence of a target word in a translated sentence given the source words using a discriminative approach. Similar approaches operating use a multilayer perceptron [12] or a bilingual neural network to learn abstract word representations and features in order to predict word, stem and suffix translations for source words given the source context [13].

An approach that integrates discriminative classifier predictions based on context and POS tags into decoding is presented in [14].

There is limited research on modeling anaphora resolution for the translation of pronouns in a statistical machine translation system. The first approaches integrate the output of explicit anaphora resolution components within the MT system [15] often focusing in particular on neuter pronouns [16, 17, 18] with limited improvements. [19] perform a classification using automatic anaphora resolution output, discriminating between the possible French pronouns in the translation. Their neural network approach surpasses maximum entropy classification and can even be extended to perform latent anaphora resolution and translation prediction jointly. An approach for translation of the English pronoun *it* into Czech is modeled by classification of the pronoun into one of three classes triggering different treatment in the tree-to-tree-based machine translation system [20].

Since the success of machine translation depends to a great deal on the morphological complexity of the target language [21], modeling target morphology in various ways is a popular direction of research. There are many approaches that perform a prediction of the inflected word forms in the target language. Conditional random fields are a popular approach to sequence labeling which is applied to predict morphemes [22], morphosyntactic properties [23], or inflection [24, 25, 26] on the target side. The latter perform a two-step translation process, first translating into stemmed forms or lemmas and then predicting the fully inflected forms. Two-step translation into Czech [27] applies two translation systems sequentially, to translate first into simplified Czech and then into fully inflected Czech. Factored translation models treat word, lemma, part-of-speech and morphological features as separate factors and perform morphological generation in a phrase-based machine translation system [28]. Enriching the source language with linguistic information in order to address noun phrase and subject-verb agreement [29]. and using fixed-length suffixes in order to improve grammaticality of the translation output [30] are further applications of the factored model.

The presented approach is modeled based on the idea of the discriminative word lexicon [11, 31], however operating on the source side instead of the target side and predicting translations given source side features. In contrast to approaches operating on the phrase level, we model predictions for individual words, however taking a up to 6 neighboring words into account and therefore covering longer context than included in the average phrase length. The approach is closely related to [14], but differs by modeling predictions for words instead of phrases, which are less sparse and therefore should provide better estimates. In addition, we include dependency features which can cover longer distances and more implicit dependencies in the sentence.

3. Source Discriminative Word Lexicon

We implement the translation disambiguation as a prediction task. The prediction is motivated by the discriminative word lexicon [31]. While the discriminative word lexicon (DWL) operates on the target side and learns to predict for each target word whether it should occur in a given target sentence, the source discriminative word lexicon (SDWL) operates on the source side. For every source word a classifier is trained to predict its translation in the given sentence. We perform a multi-class classification task by identifying for every source word the 20 most frequent translations as provided by the word alignment generated with GIZA++. All target language words that occur less often than the 20 most frequent words are assigned to one class, called **other**. Alignments to the NULL word on the target side are treated in the same way as if NULL were a word. We limit the source vocabulary to the words occurring in the test data and train up to 20 classifiers for each source word. In reality, most words have a lot less than 20 alternative translation options. The SDWL uses binary maximum entropy classifiers trained using the one-against-all scheme. That means we use a maximum entropy model to estimate $p(e|f, c(f))$, where e is the target word we want to predict given source word f and its context/dependency features $c(f)$. During training the maximum entropy models for the individual classes for each source word are learned based on the given set of features extracted from the source sentence and the correct class of each training example. For the prediction, the test data is first separated into words. For each word the features are extracted from the source sentence it stems from. Then all the binary maximum entropy models for the multiple classes are applied and each of them produces a prediction. The final prediction corresponds to the class with the highest prediction probability.

3.1. Structural Features

The training examples and test data for the classifiers are represented by a set of features and the class this example belongs to. We experiment with different types of features representing the structure of a sentence to varying degrees.

3.1.1. Bag-of-Words

A straight forward way to represent the source sentence for this classification task is to use the bag-of-words approach. This is the least structural informative feature which does not provide any knowledge about the sentence beyond the mere existence of the words in it.

3.1.2. Context

The context feature adds structural information about the local context of the modeled source word in the sentence. In addition to the context words themselves, their position is encoded in the feature such that the same word occurring at

a different position (relative to the source word in question) would result in a different feature. We include up to six context words, three on each side of the source word. Hence, this feature type provides structural information by means of sequential order within a limited context.

3.1.3. Dependency Relations

The feature contributing the most information about the sentence structure is based on the relations between the source sentence words in a dependency tree. In order to obtain the dependency relations, we extract a dependency tree from a constituency parse tree using the Stanford Parser [32, 33]. Then we include the dependency relations between the source word and its parent and children in the dependency tree as features. That means, we form a feature consisting of the governance relation (parent or child of the source word), the dependency relation type (from the set of dependency relations described in [34] e. g., nsubj, dobj, vmod, ...) and the connected word itself. This type of feature allows to capture structure by means of semantic dependencies that can range over longer distances in the sentence, but are relevant due to the semantic connection to the current source word. An example for the features for the word *it* in a given sentence is presented in Example 3.1.

Sentence:		*Well it obviously is not.*
Features:	bag-of-words	not is it obviously well .
	context	-1_well $+1$_obviously $+2$_is
	dependency	dep_parent_nsubj_is

Example 3.1: *Representation of the source word "it" by the different features*

3.2. Word Representation

We compare two methods to represent the words in the features: word IDs and word vectors.

3.2.1. Word IDs

When representing words by word IDs, we use the source vocabulary size V_{source} as the dimension of the feature space, a word's ID in the vocabulary as a feature and we set the feature to 1 if it is used in the example. All other features are set to 0. For accommodating the context features (**context**), we extend the size of the features space such that $V_{context} = c * V_{source}$ where c equals the size of the context. Each position of a word in the context hence has its own range in the features space, and words in different context positions can be distinguished accordingly. The features representing dependency relations (**dep**) are included in a similar fashion. Again, a new feature space is defined as $V_{dep} = d * V_{source}$ where d equals the amount of all dependency relations, where parent and child relations are counted separately. The feature types can be combined by simply concatenating the individual feature spaces. That means when all three types of features are used the size of the feature space amounts to $V_{source} + V_{context} + V_{dep}$. It is obvious, that with this strategy for design the feature space grows quite big, possibly leading to data sparseness problems. In order to reduce dimensions, the representation via word vectors seemed an appropriate measure.

3.2.2. Word Vectors

The word vectors for feature representation are generated using word2vec [35] with the number of dimensions set to 100. That means each word is represented by a 100-dimensional vector. However, it is not straight forward how multiple words should be expressed in this representation, so that the representation by word vectors is not applied for the bag-of-words features, but only for the context and dependency features. In case of the vector representation of the context features (**contextVec**), each position in the context words receives its own range in the feature space. Hence, the size of the feature space equals to $V_{contextVec} = c * dim$, where c is the context size and dim the dimension of the vector representation. This amounts to a significant reduction compared to $V_{context}$ used in the representation method via word IDs. The feature space for dependency relations using word vectors (**depVec**) equals to $V_{depVec} = d * dim$ with d being the inventory of dependency relations. Compared to V_{dep}, again a huge reduction. In addition to the **depVec** feature, further variants of the dependency feature are compared:

parentDepVec
: For this feature, only the dependency relation to the parent word is represented in vector representation.

parentWordVec
: This feature consists of the vector representation of the parent word and an additional binary feature that is 1 if the parent word is the root of the dependency tree.

parentWordVec+DepRel
: In addition to the **parentWordVec** feature, the dependency relation to the parent word is encoded as a vector.

As for the word-based features, word vector features can be combined by concatenation of feature spaces.

3.3. Integration of SDWL Predictions

In order to integrate the individual translation predictions into a machine translation system we use the prediction probabilities for individual words to produce scores for whole sentences. The combination of individual translation predictions for words into a sentence score is explained in the following. These scores are then used in N-best list re-ranking.

3.3.1. SDWL-based Re-ranking Scores

For each of the translation hypotheses in the N-best list, we generate a sentence score based on the translation predictions for the individual words in the sentence. We compare four methods to combine the individual word scores into a sentence score for a particular translation hypothesis.

Absolute number of predicted words
: We count the number of word translations in the sentence that coincide with the predicted translations by the translation prediction model.

Relative number of predicted words
: As an alternative score we again count the number of words in the translation hypothesis that coincide with the predicted translation. This number of matches is then divided by the total number of target words generated by the source words according to the alignment.

Sum of prediction probabilities
: For every source word we sum up the prediction probabilities associated with their aligned words in the hypothesis.

Sum of prediction ranks
: Instead of summing up the prediction probabilities of the words in the hypothesis, we sum up the ranks of the words according to their prediction probability.

All these scores were both used individually and collectively as additional sentence scores for N-best list re-ranking, in order to find out which of them are most beneficial for judging translation quality.

4. Experiments

We perform two types of experiments with the presented source discriminative word lexicon. First we use it independently to predict the translation for individual source words in the sentence and measure the prediction accuracy against the reference translation. Afterwards, we combine the individual predictions for words into a sentence score and use it in N-best list re-ranking of machine translation output.

4.1. Data

We train the classifiers on the parallel training data consisting of TED talks provided for the IWSLT 2014 evaluation campaign. Due to the limitation of the source vocabulary to the test data, we train 26,498 classifiers for 5,389 source words, which equals to an average of 4.9 translation alternatives per word. The prediction accuracy of the source discriminative word lexicon is measured on test2011 and test2012 combined. The impact of the source discriminative word lexicon on translation quality is measured after N-best list re-ranking the output of a machine translation system with the SDWL sentence score. The translation system is tuned on test2011 and tested on test2012. For N-best list re-ranking the three data sets test2010, dev2010, and test2011 are used. Translation quality before and after rescoring is reported on test2012.

4.2. Translation System

The re-ranking experiment is done using a phrase-based machine translation system. The phrase table is built using the Moses toolkit [36] and n-gram language models are trained with the SRILM toolkit [37]. Translations are generated with a phrase-based MT decoder [38]. Optimization is done with a variant of MERT [39]. Translation quality is measured in BLEU [40].

In addition to the basic translation model and language model, the system applies several word-based, POS-based and cluster-based language models, as well as a bilingual language model. Furthermore, an original discriminative word lexicon for the target side is included. Several word reordering models are used. Tree-based and POS-based reordering rules produce reordering variants of each source sentence stored in a word lattice and a lexicalized reordering model provides probability scores for the order of phrases in the translation hypotheses produced by the decoder. The translation system is described in detail in [41]. In addition, the SDWL in reduced form using only bag-of-words and context features is applied in three other systems. A German-English News system [42], an English-German and German-English TED system [43].

4.3. N-best List Re-Ranking

As mentioned above the predictions from the SDWL are combined into sentence scores for the translation hypotheses in the N-best list produced by the translation system. Then N-best list re-ranking is performed as described in [41] using the ListNet algorithm [44].

5. Results

This section presents the results of the translation prediction model tested on English-to-German translation of TED talks. First, we will show that the prediction accuracy improves when applying the proposed set of structural features. In addition, the translation quality can be improved when using the translation predictions for N-best list re-ranking to find a better translation among the hypotheses in the N-best list of the translation system.

5.1. Translation Prediction

We compare the different features for representing the sentence and context for the translation prediction of individual source words described above. We measure the accuracy of the translation prediction achieved with each of the features and feature combinations. Table 1 presents an overview of the experiments. It shows the average prediction accuracy on all words in the data used for testing.

The baseline prediction is performed with a maximum likelihood classifier, which a priori chooses the most frequent class, without using any features at all. We can see that using the bag-of-words features consisting of the words contained in the source sentence already improves over the baseline prediction. When applying the more structurally informative features, both context and dependency features individually improve considerably over the simple bag-of-words features. Among the context feature variants, the vector representation with 2 words of context in both directions performs best. For the dependency features, it is the vector representation using both parent and child relations, which leads to the best predictions. Combining the two best performing features **contextVec** and **depVec**, holds another small improvement leading to a prediction accuracy that is more than 7% higher than the baseline prediction, which corresponds to 14% relative improvement.

	Prediction Accuracy
Baseline	52.09
Bag-of-Words	53.29
Context (+/- 2 words)	58.74
ContextVec (+/- 2 words)	58.97
ContextVec (+/- 3 words)	57.48
Dep	56.07
DepVec	57.27
ParentDepVec	55.02
ParentWordVec	54.65
ParentWordVec+DepRel	55.20
ContextVec (+/-2) + DepVec	59.37

Table 1: *Translation prediction results: all words*

5.1.1. Pronoun Translation

In order to explicitly measure the accuracy of the translation prediction for pronouns, we selected the pronouns among the source words and measured the prediction accuracy of those words. Table 2 presents the prediction accuracy of source language pronouns. The pronouns achieve higher absolute numbers of translation accuracy. However, the improvements by the different types of features is comparable to the improvements on all words. The use of structural features led to an absolute and relative increase in prediction accuracy by more than 5% and 9%, respectively.

5.1.2. Subject-Verb Agreement

We also analyzed the accuracy of prediction features with respect to subject-verb agreement. For this purpose all word pairs connected by a subject relation were extracted from the dependency trees for the source sentences. All words posing as parents in such a dependency relation were taken to be possible verbs, and all children in a subject relation are considered as possible subjects. It has to be noted, though, that

	Prediction Accuracy	
	all words	pronouns
Baseline	52.09	59.58
Bag-of-Words	53.29	60.03
ContextVec (+/- 2 words)	58.97	64.89
DepVec	57.27	63.12
ContextVec (+/-2) + DepVec	59.37	65.08

Table 2: *Translation prediction results: pronouns*

the subject and verb list can also contain words of other parts-of-speech, since relations such as the one between nouns and adjectives can also be defined as a subjective relation in a dependency tree. However, manual inspection confirmed that apart from a few outliers it was indeed mostly words qualifying as subjects and verbs in the extracted list and we chose not to apply an additional manual filter. In order to produce comparable results, we measured the prediction accuracy of the words in the subject and verb lists in the same way as all words and pronouns in the results reported above. The results are presented in Table 3. We can see that the improvements of subjects and verbs are even higher than the ones on pronouns or all words, getting as close as 10% absolute and 20% relative over the baseline prediction.

	Prediction Accuracy		
	all words	subj.	verbs
Baseline	52.09	46.81	46.71
ContextVec (+/-2) + DepVec	59.37	56.00	54.12

Table 3: *Translation prediction results: subjects and verbs*

5.2. N-Best List Re-ranking

The results of improved prediction accuracy of the SDWL model with structural informative features presented above are encouraging. Therefore, we want to use the predictions to judge the quality of a particular translation hypothesis in N-best list re-ranking. For the baseline, an N-best list re-ranking is performed, using the original sentence-based scores available from the translation system. Then we compare the four ways of generating an additional score for a given hypothesis based on the individual word translation predictions described above: absolute and relative number of predicted words in the hypothesis, sum of the prediction probabilities of the words chosen in the hypothesis and rank of the words in the hypothesis according to prediction probabilities. We use the SDWL features that performed best in the previous experiment, i.e. the context vectors with context +/-2 words and the dependency vectors.

Table 4 shows an overview over the results. Three of the methods to create the sentence score perform very similar, providing about 0.2 BLEU points of improvement. Only

Source: *I memorized in my anatomy class the origins and exertions of every muscle [...]*
Baseline: *Ich in meinem Anatomie der Klasse die Ursprünge und Strapazen eines jeden Muskel [...] auswendig [...]*
+SDWL: *Ich in meiner Klasse Anatomie die Ursprünge und Strapazen jeder Muskel [...] auswendig [...]*
Reference: *In meiner Anatomievorlesung lernte ich die Ursprünge und Ausläufer jedes Muskels [...]*

Example 5.1: *Correct gender for pronoun*

Source: *There I think that the arts and film can perhaps fill the gap, and simulation.*
Baseline: *Ich glaube, dass die Kunst und Film kann vielleicht die Lücke füllen, und Simulation.*
+SDWL: *Ich glaube, dass die Kunst und Film, vielleicht können die Lücke füllen, und Simulation.*
Reference: *Hier können, denke ich, die Kunst und der Film vielleicht die Lücke füllen, sowie Simulationen.*

Example 5.2: *Correct case agreement between subject and verb*

when using the prediction ranks of the words in the hypothesis, the translation quality is not increased. That means that the translation predictions can indeed serve as an indicator for translation quality when combined in one of the three proposed ways. By using the SDWL-based scores it is possible to select an even better hypothesis from the N-best list compared to using only the available scores from the translation system.

Translation System	TED (2014) EN-DE
Baseline	24.04
SDWL: Abs	24.20
SDWL: Rel	24.22
SDWL: Sum	24.21
SDWL: Rank	23.98

Table 4: *Prediction features in re-ranking: EN-DE TED*

5.2.1. Additional Systems

The SDWL was further applied in several other translation systems in recent evaluation campaigns. Due to time constraints only the context features consisting of +/-3 words were used for the translation prediction. Table 5 shows the improvements that were gained from N-best list re-ranking with the SDWL on German-English translation of News in the WMT 2015 shared translation task as well as German-English and English-German translation of TED talks in the IWSLT 2015 machine translation task. Depending on the language and the task between 0.3 and 0.6 BLEU points can be gained from including the translation predictions even when using only the surrounding 3 context words.

Translation System	News	TED (2015)	
	DE-EN	DE-EN	EN-DE
Baseline	27.87	29.59	26.36
SDWL (ContextVec +/-3)	28.18	29.87	26.90

Table 5: *Prediction features in re-ranking: additional results*

5.3. Translation Examples

Example 5.1 shows an improvement in pronoun translation that was achieved with the SDWL. In this translation the baseline translation produces the pronoun where the gender is incorrect. Within the prepositional phrase the gender of the possessive pronoun needs to agree with its associated noun *Klasse*, which is feminine. When using the SDWL the correct gender is generated in the translation.

Example 5.2 shows that the translation prediction model also encourages morphological agreement between subject and verb. Since the information that the verb is actually in plural form is not encoded in the source language (The English verb *can*, can be both singular and plural), rendering a plural verb in the translation is not straight forward. Hence, the structural features are able to capture the plural subject in the dependency feature and/or the plural indicator *and* in the context feature, and rank the hypothesis higher where the plural verb (*können*) occurs in the translation.

6. Conclusions

We have presented a model for translation disambiguation using structural features in a classification task. The translation of a source word in a given sentence is predicted based on the classification into one of its 20 most frequent translation options. Structural features such as source context words and relations in the dependency tree of the source sentence allow to include knowledge about the sentence structure when modeling the prediction. The model is in particular aimed at improving challenging linguistic issues like the translation of pronouns and generating morphological agreement in the translated sentence.

The prediction results have shown that the accuracy of predicting a translation for individual source words increases considerably when including the context and dependency features. Representing the features by a word2vec word vector representation both reduces dimensions and increases prediction accuracy. Even though the context and dependency features contribute similar improvements individually, their combination provides the highest prediction accuracy. A separate inspection of pronouns, subjects and verbs con-

firms that these were improved in particular by up to 10%.

The individual translation predictions for the source words in each sentence are combined into a sentence score used in N-best list re-ranking. Using the prediction scores in re-ranking provides between 0.2 and 0.6 BLEU points of improvement.

Directions for future work could be the investigation of features that include more semantic information such as the semantic distance between words, or the replacement of the current classification approach by other machine learning techniques such as neural networks which are able to model more implicit dependencies. Furthermore, we would expect a positive effect on the phrase selection, if the predictions were made available already at decoding time.

7. Acknowledgements

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 645452.

8. References

[1] J. Weiner, "Pronominal Anaphora in Machine Translation," Master's thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2014.

[2] T. Herrmann, "Linguistic Structure in Statistical Machine Translation," Ph.D. dissertation, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2015.

[3] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer, "Word-sense Disambiguation Using Statistical Methods," in *Proceedings of ACL 1991*, Berkeley, CA, USA, 1991.

[4] M. Carpuat and D. Wu, "Improving Statistical Machine Translation using Word Sense Disambiguation," in *Proceedings of EMNLP-CoNLL 2007*, Prague, Czech Republic, 2007.

[5] Y. S. Chan, H. T. Ng, and D. Chiang, "Word Sense Disambiguation Improves Statistical Machine Translation," in *Proceedings of ACL 2007*, Prague, Czech Republic, 2007.

[6] L. Specia, B. Sankaran, and M. Graças Volpe Nunes, "n-Best Reranking for the Efficient Integration of Word Sense Disambiguation and Statistical Machine Translation," *Lecture Notes in Computer Science*, vol. 4919, 2008.

[7] A. Max, R. Makhloufik, and P. Langlais, "Explorations in Using Grammatical Dependencies for Contextual Phrase Translation Disambiguation," in *Proceedings of EAMT 2008*, Hamburg, Germany, 2008.

[8] K. Gimpel and N. A. Smith, "Rich Source-side Context for Statistical Machine Translation," in *Proceedings of WMT 2008*, Columbus, OH, USA, 2008.

[9] L. Shen, J. Xu, B. Zhang, S. Matsoukas, and R. Weischedel, "Effective Use of Linguistic and Contextual Information for Statistical Machine Translation," in *Proceedings of EMNLP 2009*, Suntec, Singapore, 2009.

[10] R. Haque, S. K. Naskar, A. van den Bosch, and A. Way, "Integrating Source-language Context into Phrase-based Statistical Machine Translation." *Machine Translation*, vol. 25, no. 3, 2011.

[11] A. Mauser, S. Hasan, and H. Ney, "Extending Statistical Machine Translation with Discriminative and Trigger-based Lexicon Models," in *Proceedings of EMNLP 2009*, Suntec, Singapore, 2009.

[12] A. Patry and P. Langlais, "Prediction of Words in Statistical Machine Translation using a Multilayer Perceptron," in *Proceedings of MT Summit XII*, 2009.

[13] K. M. Tran, A. Bisazza, and C. Monz, "Word Translation Prediction for Morphologically Rich Languages with Bilingual Neural Networks," in *Proceedings of EMNLP 2014*, Doha, Qatar, 2014.

[14] A. Tamchyna, F. Braune, A. M. Fraser, M. Carpuat, H. D. III, and C. Quirk, "Integrating a Discriminative Classifier into Phrase-based and Hierarchical Decoding," *Prague Bull. Math. Linguistics*, vol. 101, pp. 29–42, 2014.

[15] R. Mitkov, S. kwon Choi R, and A. Sharp, "Anaphora resolution in Machine Translation," in *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, Leuven, Belgium, 1995.

[16] R. Le Nagard and P. Koehn, "Aiding Pronoun Translation with Co-reference Resolution," in *Proceedings of WMT 2010*, Uppsala, Sweden, 2010.

[17] L. Guillou, "Improving Pronoun Translation for Statistical Machine Translation," in *Proceedings of the Student Research Workshop at EACL 2012*, Avignon, France, 2012.

[18] C. Hardmeier and M. Federico, "Modelling Pronominal Anaphora in Statistical Machine Translation," in *Proceedings of IWSLT 2010*, Paris, France, 2010.

[19] C. Hardmeier, J. Tiedemann, and J. Nivre, "Latent Anaphora Resolution for Cross-Lingual Pronoun Prediction." in *Proceedings of EMNLP 2013*, Seattle, WA, USA, 2013.

[20] M. Novák, A. Nedoluzhko, and Z. Žabokrtský, "Translation of "It" in a Deep Syntax Framework," in *Proceedings of DiscoMT 2013*, Sofia, Bulgaria, 2013.

[21] A. Birch, M. Osborne, and P. Koehn, "Predicting Success in Machine Translation," in *Proceedings of EMNLP 2008*, Honolulu, HI, USA, 2008.

[22] A. Clifton and A. Sarkar, "Combining Morpheme-based Machine Translation with Post-processing Morpheme Prediction," in *Proceedings of ACL-HLT 2011*, Portland, OR, USA, 2011.

[23] S. Green and J. DeNero, "A Class-based Agreement Model for Generating Accurately Inflected Translations," in *Proceedings of ACL 2012*, Jeju, South Korea, 2012.

[24] K. Toutanova, H. Suzuki, and A. Ruopp, "Applying Morphology Generation Models to Machine Translation," in *Proceedings of ACL-HLT 2008*, Columbus, OH, USA, 2008.

[25] A. Fraser, M. Weller, A. Cahill, and F. Cap, "Modeling Inflection and Word-Formation in SMT," in *Proceedings of EACL 2012*, Avignon, France, 2012.

[26] A. E. Kholy and N. Habash, "Translate, Predict or Generate: Modeling Rich Morphology in Statistical Machine Translation," in *Proceedings of EAMT 2012*, Trento, Italy, 2012.

[27] D. Mareček, R. Rosa, P. Galuščáková, and O. Bojar, "Two-step Translation with Grammatical Post-processing," in *Proceedings of WMT 2011*, Edinburgh, Scotland, 2011.

[28] P. Koehn and H. Hoang, "Factored translation models," in *Proceedings of EMNLP-CoNLL 2007*, Prague, Czech Republic, 2007.

[29] E. Avramidis and P. Koehn, "Enriching Morphologically Poor Languages for Statistical Machine Translation," in *Proceedings of ACL-HLT 2008*, Columbus, OH, USA, 2008.

[30] N. S. Razavian and S. Vogel, "Fixed Length Word Suffix for Factored Statistical Machine Translation," in *Proceedings of ACL 2010*, Uppsala, Sweden, 2010.

[31] J. Niehues and A. Waibel, "An MT Error-driven Discriminative Word Lexicon using Sentence Structure Features," in *Proceedings of WMT 2013*, Sofia, Bulgaria, 2013.

[32] D. Klein and C. D. Manning, "Fast Exact Inference with a Factored Model for Natural Language Parsing," in *Proceedings of NIPS 2002*, Vancouver, Canada, 2002.

[33] ——, "Accurate Unlexicalized Parsing," in *Proceedings of ACL 2003*, Sapporo, Japan, 2003.

[34] M.-C. de Marneffe and C. D. Manning, "Stanford typed dependencies manual," Stanford University, Stanford, CA, USA, Tech. Rep., 2008.

[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *CoRR*, vol. abs/1301.3781, 2013.

[36] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of ACL 2007*, Prague, Czech Republic, 2007.

[37] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *Proceedings of ICSLP 2002*, Denver, CO, USA, 2002.

[38] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[39] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *Proceedings of WPT 2005*, Ann Arbor, MI, USA, 2005.

[40] 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.

[41] 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 IWSLT 2014*, Lake Tahoe, USA, December 2014.

[42] E. Cho, T.-L. Ha, J. Niehues, T. Herrmann, M. Mediani, Y. Zhang, and A. Waibel, "The Karlsruhe Institute of Technology Translation Systems for the WMT 2015," in *Proceedings of WMT 2015*, Lisbon, Portugal, September 2015.

[43] T.-L. Ha, J. Niehues, E. Cho, M. Mediani, and A. Waibel, "The KIT Translation Systems for IWSLT 2015," in *Proceedings of IWSLT 2015*, Da Nang, Vietnam, December 2015.

[44] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to Rank: From Pairwise Approach to Listwise Approach," in *Proceedings of ICML 2007*, Corvalis, OR, USA, 2007.

Phrase-level Quality Estimation for Machine Translation

Varvara Logacheva, Lucia Specia

Department of Computer Science
University of Sheffield, UK
v.logacheva@sheffield.ac.uk, l.specia@sheffield.ac.uk

Abstract

The paper presents the first attempt to perform quality estimation (QE) of machine translation (MT) at the level of phrases. Automatically translated sentences directly or indirectly labelled by humans for quality at the word level are used to devise phrase-level quality labels. We suggest methods of segmenting sentences into phrases which mimic the actual segmentation that generated the translations. For the prediction models, we apply two sets of phrase-level features: (1) features used in sentence-level QE work, (2) features based on word vector representations. Our experiments show that the phrase-level models can improve over word-level models in terms of how well they detect errors.

1. Introduction

Quality estimation (QE) of machine translation (MT) aims at determining the quality of an automatically translated text without comparing it to a reference translation. This task often arises in real-world applications of MT, e.g. when users of an MT system translate new data and are interested in understanding how reliable the system output is. No reference translations are available for such data, and therefore the use of standard MT evaluation metrics is not possible. The only way of determining the quality of the automatic translation is the use of indirect evidence. QE is particularly useful in applications which provide automatic translations for gisting and in computer-assisted translation (CAT) settings where automatic translation is followed by post-editing by humans.

The QE task started as the estimation of confidence of individual words in a translated sentence with respect to a particular translation model. Back then the task focused on the confidence of a particular MT system about an automatic translation, and as such explored features that required information from the MT system, such as hypotheses and n-best lists statistics [1], word posterior probabilities [2], n-gram posterior probabilities [3].

More recently QE acquired a broader sense [4]: estimating the quality of a translation for a particular purpose (e.g. gisting or further post-editing), often disregarding the MT system that generated it. The features currently used in QE are thus system-independent; they use properties of the source text and its translation (e.g. number of tokens, numbers, punctuation marks in sentences) or information from external resources not related to the MT system that produced the translation (POS tags, syntactic features, perplexity under external LMs) [5].

The labelling of translations (and therefore the score to estimate) has changed as well: instead of using automatic MT evaluation metrics to produce labels, the labelling is more often done by humans (e.g. post-editing effort of a sentence within to a 1-5 point scale [4]) or deduced from manually generated data (e.g. post-editing effort defined by the percentage of editing a translator performed, or post-editing time measured by a CAT tool [6]). These are all labels for sentence-level QE. Word-level labels, on the other hand, are less clearly defined.

The task of word-level QE has regained attention since 2013, when it became part of the WMT evaluation campaign [6]. The post-editing of MT output was used to automatically collect translations annotated for quality at the word level: a word left unchanged by a translator was labelled as "OK", while a word edited was labelled as "BAD". However, framing the QE task in this way has serious limitations. Notably, the fact that errors in different words are not independent from one another. For example, if two words agree in their grammatical features, changing one of them will most likely cause the need to change the other one as well. For example, if we translate the English phrase "My dear friend" into French, a possible translation is "Mon cher ami". However, a post-editor will change it into "Ma chère amie" if "friend" refers to a feminine entity. Here one mistranslation ("ami" instead of "amie") will have resulted in three corrections.

Such groups of related edits were defined in [7] as post-editing actions (PEAs) — minimal units that should be post-edited jointly in one action according to some pattern. The MQM (Multidimensional Quality Metrics) framework [8] for translation error analysis also focuses on defining errors that can span phrases of any length. This leads us to the idea that QE should be done at the level of phrases, as opposed to words. Analysing groups of words jointly can provide additional information which is not available at the word level, and notifying a user that the errors in several adjacent words are related can help them use quality predictions more efficiently.

Another motivation for phrase-level QE is the fact that the most widely used MT engines are phrase-based, i.e. at each step the MT decoder extends the translation hypothe-

sis with a phrase. In other words, decisions are made over phrases, rather than over single words. Therefore, it is likely that translation errors can also be generated at the phrase-level. In addition, phrase-level QE models could be used to guide decoding to avoid certain errors.

Previous work on word-level QE has highlighted the intuition that errors can span over entire phrases. [9] use a number of features that rely on the source phrase that generated the current target phrase. In [2] the word posterior probability is computed at the phrase level: it is regarded as the probability of a word being generated by a source phrase rather than by the entire source sentence. However, in previous research the quality labels are defined for every word, and thus our work represents the first effort to estimate the quality of a target phrase as an atomic unit. We identify the main challenges in this task and suggest ways of dealing with them.

The biggest challenge in phrase-level QE is segmentation: the task requires the automatic translations to be segmented into phrases, and each phrase to be labelled for quality. Although there exist datasets labelled for errors at the phrase level (e.g. using the MQM framework [10]), they do not provide a segmentation that can be used directly for the task. Since only errors are labelled, very long sequences of error-free segments are found in these datasets, and there is no clear way to segment them. If we train a classifier based on such data to discriminate between good and bad phrases, it is very likely to be biased by a phrase length and to classify shorter phrases as bad and longer phrases as good regardless of their actual quality. In addition, if the phrase segmentation is done based on the reference labels, we have no way of segmenting unseen data, for example the test data to evaluate the model's performance.

Since no existing phrase-labelled datasets can be used for the task, we explore and adapt datasets labelled for quality at the word level. We expand this labelling by performing decoder-like segmentation. We test different sets of features and compare the performance of phrase-level QE models on different feature sets.

The rest of the paper is organised as follows. In Section 2 we describe our segmentation strategies and ways of adapting word-level labels for phrases. Sections 3 and 4 describe the feature sets and training algorithms and in Section 5 we report the results of our experiments.

2. Segmentation and labelling

Phrase-level QE relies heavily upon appropriate sentence segmentation. One of the main difficulties involved in the segmentation task is the lack of a strict definition of what a *phrase* is for this purpose. In linguistics, *phrase* is a unit where words are connected by dependency relationships. In statistical MT, phrases are simply sequences of words that frequently co-occur and are aligned with the same source word sequences.

Given that a lot of the translation data is likely to be produced by statistical MT systems nowadays, for this work we assume the latter notion of segmentation and reproduce the segmentation produced by a statistical MT decoder. Since we do not have access to the MT system that produced the translations, we re-decode the source data with a statistical MT system and reproduce its phrase segmentation. We are not guaranteed that this segmentation will match the original one, i.e., the one that generated the target data. However, the two MT systems are very similar, and thus we hope to get similar segments. We suggest two ways of segmenting sentences into Moses-like phrases [11]: segmentation of both source and target sentences jointly with a source-target MT system, and independent segmentation of target sentences.

2.1. Source segmentation

The datasets we use for QE systems training have source sentences and their automatic translations. If we had access to the MT system which generated the translation, we could reproduce the original segmentation accurately by simply re-decoding the source sentences. However, such MT models are rarely made available, and we are not guaranteed to get the same output using another MT system, even if it trained on the same data.

One possible solution is to constrain the decoder to use only phrases that appear in the target sentence. However, constrained decoding is often unable to fully reach the translation provided, usually because of out-of-vocabulary (OOV) words or lack of suitable phrases in the phrase table. In order to supply the system with this information we trained an additional phrase table on the data to be decoded (i.e. phrase-level QE data), and produced translations using both phrase tables.

Despite this additional data-specific phrase table, a small percentage of sentences still could not be decoded. In those cases we considered each word of the sentence a separate phrase, and the corresponding source phrase as the word aligned to it. Therefore, for some "phrases" of such sentences, the source phrase will be empty.

2.2. Target segmentation

Our second technique consists in segmenting only the target sentence with an MT system which translates from the target language into the source language. We translate the target sentence with no constraints and retrieve the phrase segmentation for it. The actual translation will not match the source side of our data, which is not an issue as we will not use it. Moreover, we suppose that the output language of such a system is not important, because we only use it to segment the input sentence.

We obtain the source segmentation by combining the target segmentation and source-target alignments: for each target phrase, the corresponding source phrase is composed of all source words aligned to the words in the target phrase. The source phrase needs to be continuous, i.e. if two source

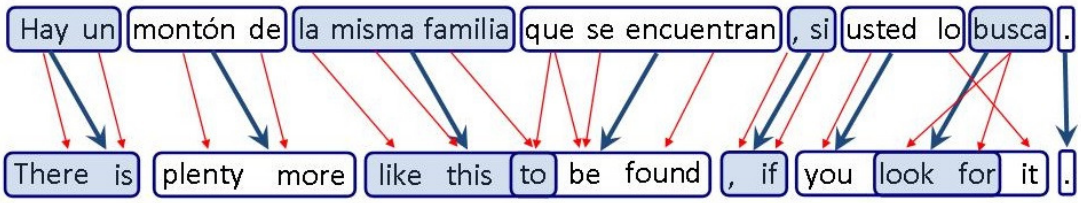


Figure 1: Overlapping source phrases generated by projection of target phrases onto the source sentence. Red lines denote word alignments, blue lines denote phrase alignments.

words aligned to one target phrase have an unaligned word (or a word aligned to another target phrase) in between, this unaligned word also has to be included into the corresponding target phrase. Figure 1 shows this case: the phrase *"usted lo"* is aligned to two source words: *"you"* and *"it"*. They have two words between them, so the source phrase will be *"you look for it"*. The figure also gives an example of overlapping source phrases.

This approach has a few drawbacks: it can include a source word in more than one phrase, it does not guarantee the full coverage of a source sentence and it can generate an empty source phrase if all words in the target phrase are unaligned. In addition to that, when performing this type of segmentation we feed the automatically translated sentences to the decoder, because the target side of our QE training data has been generated by an MT system. Since the majority of these sentences contain errors, the phrase segmentation for them can be different from the one generated for a valid target language sentence.

2.3. Phrase labelling

Datasets with post-edited machine translations can be labelled at the word level by comparing the automatic translations with its post-edited version. This can be done with edit distance metrics such as the one implemented in the Tercom tool [12]. This tool identifies an edit operation (substitution, deletion, shift) which needs to be performed on a word to make the automatic translation match its post-edition. The word labels could thus be the edit operations which need to be performed on words to improve the sentence translation, as in the dataset created for the WMT-13 QE shared task [6]. Other datasets have incorrect words manually labelled with fine-grained error classes (grammatical error, mistranslation, etc.) [10]. However, since the number of errors is relatively small (10-30% for different datasets), in order to reduce sparsity, binary ("OK"/"BAD") labels are often used [10, 13]. They indicate simply whether a word suits the context or needs to be edited. However, both these types of labels are defined over words only. When segmenting a sentence with one of the techniques described above, we are likely to face a situation where words put together into a phrase have different tags. Thus we need to combine word labels to get to a single phrase label.

The most obvious combination strategy is *majority labelling*, i.e. to assign the most common label of the words in the phrase to that phrase. However, such a strategy is likely to further increase the skewed discrepancies between the number of occurrences of "BAD" and "OK" labels. The majority tagging strategy can reduce even more the number of "BAD" tags, which will in turn make learning harder. We propose three alternative labelling strategies to mitigate this issue:

- optimistic — if half or more of words have a label "OK", the phrase has the label "OK" (majority tagging),
- pessimistic — if 30% words or more have a label "BAD", the phrase has the label "BAD",
- super-pessimistic — if any word in the phrase has a label "BAD", the whole phrase has the label "BAD".

The latter strategy is motivated by the possibility of using phrase-level QE to support phrase-based MT decoding. At each step of the search process the decoder chooses a new phrase, and the best candidate phrase should contain only "good" words. If one of the words does not fit into the context, the entire phrase should be considered unsuitable.

2.4. Joint target+data segmentation

Instead of changing the edit distance-based labels, we can get rid of phrases with ambiguous tags if we combine the phrase borders identified by the decoder with the borders of "OK" and "BAD" spans in our data. Let us consider the following example. The target phrase *"¿Sabes lo que voy a hacer, sin embargo?"* and its original edit distance-based tagging *"OK OK OK OK BAD BAD BAD BAD BAD BAD OK"* create the following segmentation:

[¿ Sabes lo que] [voy a hacer , sin embargo] [?]

The **target segmentation** procedure for the same sentence returns a different segmentation with ambiguous tags:

[¿ Sabes] [lo que voy a hacer] [,] [sin embargo] [?]

OK OK/BAD BAD BAD OK

However, if we combine two sets of borders, we convert one phrase with ambiguous tagging (*"lo que voy a hacer"* — 2 "OK", 3 "BAD" words) into two unambiguous phrases:

[¿ Sabes] [lo que] [voy a hacer] [,] [sin embargo] [?]

OK OK BAD BAD BAD OK

Note that we can join the phrase borders with the label span borders only for the target segmentation, because the source segmentation has the corresponding source phrases, which cannot be segmented into "BAD" and "OK" segments.

2.5. Evaluation

In order to implement a phrase-level QE system we need to segment both training and test data, and then label the test phrases with a trained model. However, the phrase-level output currently cannot be evaluated directly, because we have no datasets with phrase-level annotation. Therefore, we segment the test sentences into phrases and label them, and then we propagate the phrase labels onto all words of the phrase. After that the test output can be evaluated at the word level.

3. Features

The the majority of features used in word-level QE systems cannot be applied to phrases. However, most of the sentence-level features are suitable for any sequence of words, not only full sentences. For our experiments we used a list of 79 sentence features used in the QuEst QE framework [14][1]. These features are called "black-box" because they do not use the information from MT system. Some examples:

- LM features: language models (LM) score of source and target phrases under source and target LMs.
- POS features: numbers of verbs, nouns and other parts of speech in the source and the target.
- Features that indicate the number of tokens from different closed classes: numbers, alphanumeric tokens, punctuation marks.
- Average number of translations of source words.
- Average number of n-grams in different frequency quartiles.

Another set of features we use relies on source-only information, namely vector representations of words generated with `word2vec` tool[2]. Word2vec assigns every word a fixed-size vector of numbers that encodes information on the word's contexts. Therefore, similar words should have similar vectors (for a detailed description of `word2vec` see [15]). The vectors are word-level, but unlike other word-level features they can be easily combined for phrases that are longer than one word. We can use two vector operations to combine two or more vectors of the same size while keeping the dimensionality of these vectors: element-wise sum or average of the vectors. According to our preliminary experiments, systems trained on summed vectors showed higher performance than systems with averaged vectors, so in the experiments reported below we use the sum of the vectors.

[1]For the complete list of features: `http://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox`

[2]`https://code.google.com/p/word2vec/`

4. Training algorithms

Most word-level QE approaches rely on sequence labelling algorithms. One of the best-performing sequence labelling techniques is **conditional random fields** (CRF) [16], which has been used by many word-level QE systems [17, 18]. However, a CRF model might be less helpful for phrase-level QE. The errors in words may be dependent on each other, and thus the labels of neighbouring tokens can influence each other. Linear chain CRFs are well suited for modelling this type of dependency. However, in phrase-level QE the relatedness of word-level errors is already captured by the phrases. In other words, if the segmentation is accurate, it encapsulates related errors in one unit. While there are no constraints on labels of adjacent phrases (i.e., two or more OK/BAD phrases can occur consecutively), these labels are not expected to be as closely related as those in word-level QE. Therefore, we also explore a standard classifier, a **random forest** classifier [19], which showed good performance in our previous experiments on word-level QE.

5. Experiments

We performed a set of experiments to test how phrase-level systems compare to previous work on word-level QE and to find the optimal parameters for the phrase-level training. We tested performance varying the following parameters:

- Segmentation: target segmentation, source segmentation, target+data segmentation,
- Phrases labelling: optimistic, pessimistic or super-pessimistic,
- Feature set: sentence-level features from QuEst, combined `word2vec` word vectors, both sets of features,
- Models: CRF or Random forest.

We conducted our experiments on two datasets used for the QE shared tasks in 2014 and 2015, so we can compare the performance of our systems with state-of-the-art results.

5.1. Systems

The training of a phrase-level QE system was performed with the open-source QE tool `Marmot`[3]. We trained three distinct systems on three datasets:

- **phrase-wmt-14**: trained on the WMT-14 dataset labelled with error types [10].
- Two systems were trained on fractions of the WMT-15 dataset [13]. This dataset has 11,000 post-edited automatic translations. However, the majority of them contain too few errors, and QE systems trained on the full dataset tend to perform overly optimistic labellings. Therefore, following [20] we use only sentences with the highest HTER score (i.e. largest number of errors normalised by the sentence length):

[3]`https://github.com/qe-team/marmot/tree/phrase_level`

– **phrase-wmt-15-2000**: trained on the 2,000 worst sentences from WMT-15,
– **phrase-wmt-15-5000**: trained on the 5,000 worst sentences from WMT-15.

We also compare our system with the following representative systems that participated in the WMT-14 and WMT-15 QE shared tasks at the word level:

- Systems from WMT-14:
  - **Baseline-all-bad** — trivial baseline strategy that assigns the tag "BAD" to all words. No other system could beat it in terms of F_1-BAD score.
  - **FBK-UPV-UEDIN** [18] — system with features from word posterior probabilities and confusion network descriptors computed over 100,000-best translations. Tagging was done with bidirectional long short-term memory recurrent neural networks. This was the best system in WMT-14.
  - **LIG** [17] — system with 25 black-box features and was trained with CRF. It was the 3rd best system in WMT-14.
- Systems from WMT-15:
  - **Baseline** [13] — system that was used as a baseline at the WMT-15 word-level QE task.
  - **Baseline-all-bad** — the same "all-bad" strategy.
  - **UAlacant** [21] — system that used features drawn from pseudo-references (automatic translations of the source sentence) generated by different MT systems, and baseline features released for the task. Best best-performing system.
  - **Shef-word2vec** [22] — system that used word vector representations as features and performed labelling with a CRF model. This system was ranked 3rd out of 8.

5.2. Tools and datasets

Besides the training and test sets, a QE system requires various resources and tools for feature extraction:

- The word alignment model was trained on the Europarl corpus [23] using the `fast-align` tool[4].
- LM and n-gram count features were extracted using LMs trained on the Europarl corpus using SRILM[5].
- POS features were extracted with TreeTagger [24].
- The translation probability features were computed using lexical probability tables trained with Moses system [11] on the Europarl corpus.
- The word vector representations were computed with `gensim` [25] — Python implementation of `word2vec` models. The training data for the vectors is the concatenation of Europarl, News-commentary[6] and News crawl[7] corpora. The vectors are 500-dimensional.

[4] https://github.com/clab/fast_align
[5] http://www.speech.sri.com/projects/srilm/
[6] http://statmt.org/wmt15/

5.3. Segmentation properties

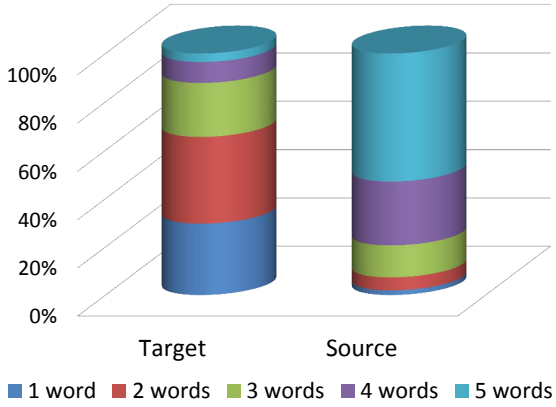


Figure 2: Phrase length frequencies for different segmentation techniques.

The segments produced by two segmentation strategies differ substantially. The main difference is the distribution of phrase lengths: while the **target** segmentation tended to segment the sentences into shorter phrases, the majority of phrases used by the **source** segmentation are 5-word long (see Figure 2). This is explained by the fact that the former strategy uses an independent translation table, whereas the latter decodes the sentences with a translation table trained on the same sentences, so it contains longer phrases.

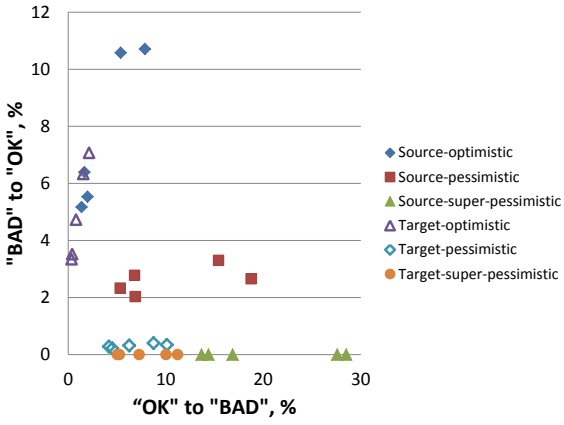


Figure 3: Percentages of word labels modified for datasets segmented with different segmentation techniques (source/target) and re-labelled with either of the labelling strategies (optimistic/pessimistic/super-pessimistic).

We also looked at the amount of word labels that were modified by different labelling strategies under the target-

[7] http://statmt.org/wmt14/

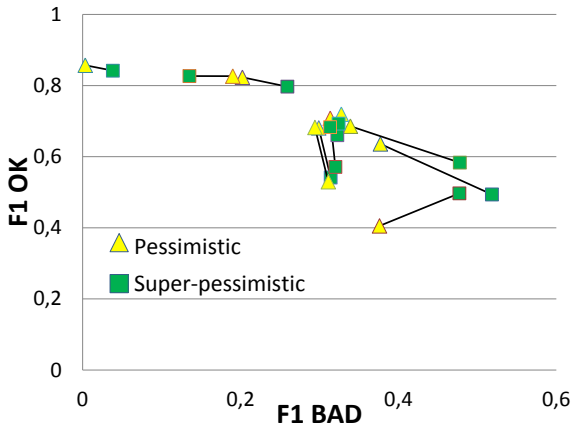


Figure 4: Results for systems with pessimistic and super-pessimistic phrase tagging schemes. Results of systems that differ only in terms of tagging strategy are joined with a line.

based and source-based segmentation types. Figure 3 shows the percentage of words in different datasets that needed to change the label from "OK" to "BAD" and vice-versa. Under source segmentation all labelling techniques become more aggressive, i.e. they change more words. The "optimistic" strategy changes zero or few words from "OK" to "BAD", whereas the "super-pessimistic" strategy does not change words from "BAD" to "OK". Datasets converted with the "pessimistic" strategy contain both types of conversions, but tend to add "BAD" labels rather than "OK" labels.

5.4. Selection of optimal parameters

Here we study which parameters we should use to achieve the best prediction quality for our datasets. We found that most of the parameters depend on datasets and values of other parameters. In addition, the performance of a system is difficult to define: as the F_1 score for the "BAD" class (primary metric for the word-level QE task used for systems comparison in [13]) grows, the F_1 score for the "OK" class drops. In order to account for both of them we plot the F_1-BAD with respect to F_1-OK scores. In each plot we compare systems that differ in one parameter. They are usually shown as items of different colours and shapes. Some items of the same configuration can lie quite far apart. That happens because other parameters of a given pair of systems influenced their performance.

The performance of systems that use different **labelling** schemes follow a certain pattern: the F_1-BAD grows as more negative data is added, while the F_1-OK score drops. Thus, the 'optimistic' labelling scheme is almost always inferior to the other two strategies. The 'pessimistic' and 'super-pessimistic' schemes perform closer, but the latter returns higher F_1-BAD scores for most settings (Figure 4).

This can also be attributed to the source **segmentation** strategy, which generates longer phrases and therefore requires more words to change tag from "OK" to "BAD". Figure 5 shows the comparison of different segmentation strategies and **training algorithms**. It can be seen that CRF produced the best- as well as the worst-performing systems depending on the type of segmentation: the source-segmented data achieves high F_1-BAD score, whereas target segmentation does not perform well in terms of F_1-BAD. On the other hand, the systems trained with the Random Forest classifier do not discriminate between the segmentation types. In addition to that, these systems proved very unstable, whereas CRF always returned the same results for a given configuration. In order to get more meaningful results, we ran the Random Forest classifier 20 times for each configuration and averaged the results.

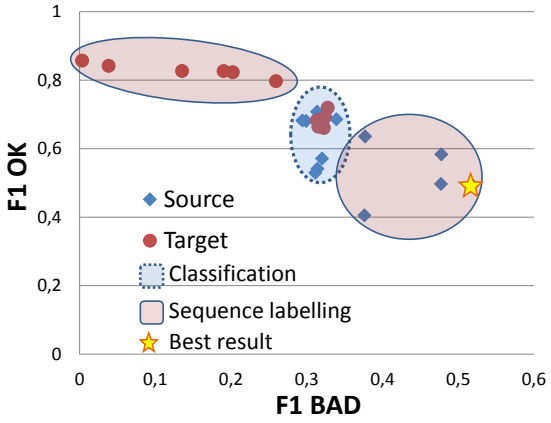


Figure 5: Differences between target and source segmentation and between classification and sequence labelling for phrase-level systems.

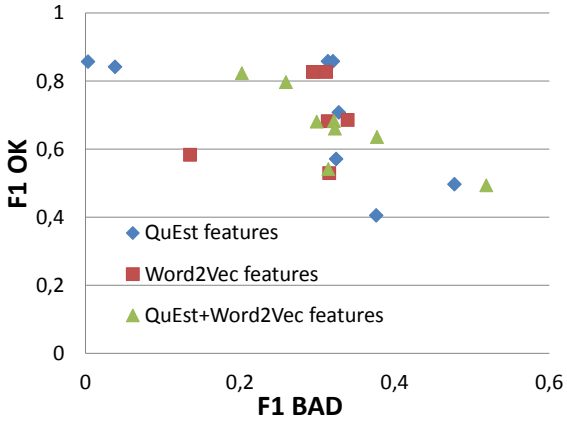


Figure 6: Performance of systems with different feature sets.

The different sets of **features** do not lead to as much variance in performance as the other parameters. However, we can notice that systems with `word2vec` features are more stable and less dependent on other parameters: all systems which use these features perform closely. The use of QuEst and `word2vec` features in combination can lead to the im-

proved performance, whereas systems using only QuEst features are the least stable.

The settings that returned the highest F_1-BAD scores for all the datasets were similar: **source segmentation**, **super-pessimistic labelling**, system trained with **CRF** (see yellow star in Figure 5). The optimal feature sets differ for different datasets. All the figures in this section show the performance of systems trained on 5,000 sentences from the WMT-15 dataset, but the the trends hold for the rest of the systems.

5.5. Comparison to word-level systems

We trained our phrase-level systems on datasets used in the WMT-14 and WMT-15 QE shared tasks, so that we can compare our systems with word-level systems for the task. The WMT-14 system used QuEst features, the WMT-15 system with 2,000 sentences — `word2vec` features, the WMT-15 system with 5,000 sentence — the combination of QuEst and `word2vec` features (although for both WMT-15 systems all feature sets performed closely). The rest of the parameters were fixed for all the datasets: source segmentation, super-pessimistic labelling, CRF.

System	F_1-**BAD** ↑	F_1-OK	Weighted F1
phrase-wmt-14	62.76	39.07	56.80
Baseline-all-bad	52.52	0.0	18.7
FBK-UPV-UEDIN	48.72	69.33	61.99
LIG	44.47	74.09	63.54

Table 1: Performance on WMT-14 test set, systems sorted from best to worst, our system in bold.

System	F_1-**BAD** ↑	F_1-OK	Weighted F1
phrase-wmt-15-5000	51.84	49.38	51.08
phrase-wmt-15-2000	51.57	49.05	50.79
UAlacant	43.12	78.07	71.47
SHEF-word2vec	38.43	71.63	65.37
Baseline-all-bad	31.75	0.0	5.99
Baseline	16.78	88.93	75.31

Table 2: Performance on WMT-15 test set, systems sorted from best to worst, our systems in bold.

Table 1 shows the performance of systems trained and tested on the QE dataset released for the WMT-14 shared task. Our system is the only system which beats the trivial all-bad baseline strategy in terms of F_1-BAD score. The same trend is seen in Table 2, which shows the performance of systems on the WMT-15 data. Both our systems outperform all other system including the winner. They achieve very close scores, which confirms that sentences with less errors do not contribute much for word-level QE.

6. Conclusions and future work

We introduced an approach for quality estimation of MT at the phrase level. To the best of our knowledge, this is the first attempt to label MT phrases with quality. We found that our phrase-level systems outperform word-level systems.

We tested a number of different parameters and found that sentence-level features give better results than word embedding features, CRF model performs better than Random Forest classifier, and the best segmentation strategy is to perform decoding of a source sentence restricting the decoder to output the target sentence, and use the phrase segmentation generated during the decoding. The best tagging strategy is to assume that every phrase that contains at least one "BAD" word should be tagged as "BAD".

In future work we will investigate the performance of other training algorithms. We believe that phrase-level QE can benefit from more advanced algorithms that take into account the segmentation of a sentence in subsequences. For example, Semi-Markov CRFs [26] are designed to solve segmentation and labelling tasks jointly, and higher order CRFs [27] explicitly consider relations between non-adjacent words which can be useful for modelling phrase errors.

An issue with phrase-level QE is that all available datasets are annotated only at the word level. Another direction for future work will thus be the development of a dataset of automatic translations annotated for quality at the level of phrases. From an application perspective, we assume that the phrase segmentation should be guided by segments in statistical MT rather than linguistic properties of the data. However, it would also be interesting to test the usefulness linguistically-informed segmentation.

Finally, further research is necessary to design features that are specific for phrase-level QE. Phrases combine properties of sentences and words: they are sequences, like sentences, but can be quite short, so sentence-level features may be uninformative. The usefulness of linguistically motivated features in particular needs to be tested: as the phrase segmentation performed by an MT decoder does not take into account linguistic information, features indicating whether a phrase is valid based on linguistic information may not suit the task. On the other hand, linguistic information can be useful as it is often unknown to the MT system.

Phrase-level QE of MT is a new field of research. In this paper we proposed the first strategy for the task, highlighted some of its challenges and outlined possible directions of future work.

7. Acknowledgements

This work was supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n° 317471.

8. References

[1] S. Gandrabur and G. Foster, "Confidence estimation for translation prediction," in *HLT-NAACL-2003*, Edmonton, Canada, 2003, pp. 95–102.

[2] N. Ueffing and H. Ney, "Word-Level Confidence Estimation for Machine Translation using Phrase-Based Translation Models," in *HLT-EMNLP-2005*, no. October, Vancouver, Canada, 2005, pp. 763–770.

[3] R. Zens and H. Ney, "N-Gram Posterior Probabilities for Statistical Machine Translation," in *WMT-2006*, no. June, 2006, pp. 72—-77.

[4] L. Specia, N. Cancedda, M. Dymetman, M. Turchi, and N. Cristianini, "Estimating the sentence-level quality of machine translation systems," in *EAMT-2009*, Barcelona, Spain, 2009.

[5] K. Shah, T. Cohn, and L. Specia, "An investigation on the effectiveness of features for translation quality estimation," in *MT Summit XIV*, Nice, France, 2013, pp. 167–174.

[6] O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2013 Workshop on Statistical Machine Translation," in *WMT-2013*, Sofia, Bulgaria, August 2013, pp. 1–44.

[7] F. Blain, J. Senellart, H. Schwenk, M. Plitt, and J. Roturier, "Qualitative Analysis of Post-Editing for High Quality Machine Translation," in *MT Summit XIII*, Xiamen, China, 2011, pp. 164–171.

[8] A. Lommel, A. Burchardt, M. Popović, K. Harris, E. Avramidis, and H. Uszkoreit, "Using a New Analytic Measure for the Annotation and Analysis of MT Errors on Real Data," in *EAMT-2014*, 2014, pp. 165–172.

[9] N. Bach, F. Huang, and Y. Al-Onaizan, "Goodness: A Method for Measuring Machine Translation Confidence," in *ACL-2011*, Portland, Oregon, 2011, pp. 211–219.

[10] C. Buck, P. Pecina, J. Leveling, M. Post, L. Specia, and H. Saint-amand, "Findings of the 2014 Workshop on Statistical Machine Translation," in *WMT-2014*, Baltimore, USA, 2014, pp. 12–58.

[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 *ACL-2007*, Prague, Czech Republic, 2007, pp. 177–180.

[12] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and R. Weischedel, in *AMTA-2006*.

[13] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, P. Koehn, C. Monz, M. Negri, P. Pecina, M. Post, C. Scarton, L. Specia, and M. Turchi, "Findings of the 2015 Workshop on Statistical Machine Translation," in *WMT-2015*, Lisbon, Portugal, 2015.

[14] L. Specia, K. Shah, J. G. C. de Souza, and T. Cohn, "QuEst - A translation quality estimation framework," in *ACL-2013*, Sofia, Bulgaria, 2013.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[16] J. Lafferty, A. Mcallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *ICML-2001*, 2001, pp. 282–289.

[17] N. Q. Luong, L. Besacier, and B. Lecouteux, "Lig system for word level qe task at wmt14," in *WMT-2014*, Baltimore, USA, June 2014, pp. 335–341.

[18] J. G. Camargo de Souza, J. González-Rubio, C. Buck, M. Turchi, and M. Negri, "Fbk-upv-uedin participation in the wmt14 quality estimation shared-task," in *WMT-2014*, Baltimore, Maryland, USA, June 2014, pp. 322–328.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] V. Logacheva, C. Hokamp, and L. Specia, "Data enhancement and selection strategies for the word-level quality estimation," in *WMT-2015*, Lisboa, Portugal, September 2015, pp. 311–316.

[21] M. Esplà-Gomis, F. Sánchez-Martínez, and M. Forcada, "Ualacant word-level machine translation quality estimation system at wmt 2015," in *WMT-2015*, Lisbon, Portugal, September 2015, pp. 309–315.

[22] K. Shah, V. Logacheva, G. Paetzold, F. Blain, D. Beck, F. Bougares, and L. Specia, "Shef-nn: Translation quality estimation with neural networks," in *WMT-2010*, Lisbon, Portugal, September 2015, pp. 342–347.

[23] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," in *MT Summit X*, 2005.

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

[25] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *LREC-2010*, Valletta, Malta, May 2010, pp. 45–50.

[26] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," *Advances in Neural Information Processing Systems 17*, pp. 1185—-1192, 2004.

[27] N. Ye, W. S. Lee, H. L. Chieu, and D. Wu, "Conditional Random Fields with High-Order Features for Sequence Labeling," *Journal of Web Semantics*, vol. 2, p. 2, 2004.

Improving Continuous Space Language Models using Auxiliary Features

Walid Aransa, Holger Schwenk, Loïc Barrault

LIUM, University of Le Mans, France

firstname.lastname@lium.univ-lemans.fr

Abstract

In this paper we introduce a novel method to improve the continuous space language models using auxiliary features. The suggested auxiliary features include text genre, line length, various types of context vector representations. We report perplexity improvements of around 7.5% of the English Penn Treebank data set. We also report an improvement on a translation task up to 1.1 BLEU point on test by re-scoring the n-best list generated by our phrase-based statistical machine translation system.

1. Introduction

The neural network LM (also known as continuous space LM or CSLM) tries to overcome the disadvantages of back-off n-gram LMs. One of these disadvantages is that the probabilities are estimated in a discrete space which does not allow directly the estimation of non-observed n-gram in the training data. In a neural network LM, the words are projected into a continuous space during the training. [1] proposes a multi-layer neural network model that jointly learns the word projection and the probability estimation. The basic architecture of this neural network is shown in Figure 1.

A CSLM has many advantages, it can be used to estimate the probability of long n-gram (also short n-gram) which can not be directly estimated using n-gram back-off LMs. Also, it can be trained using longer context with just small increase in the complexity which is not possible for n-gram back-off LMs.

The CSLM was successfully applied to large vocabulary speech recognition. It is usually used to rescore lattices and improvement of the word error rate by about one point were obtained for many languages and domains, for instance [3, 4, 5, 6]. More recently, the CSLM was also successfully applied to statistical machine translation [7, 8, 9, 10].

In this paper, we present improvements of the CSLM. The idea is to provide additional information at the input of the neural network in a similar for recurrent NN LM by [18]. We call these additional inputs "auxiliary features". We use different types of auxiliary features including line length, text genre, line context vector representation,... etc. By these means, better domain and context specific LM estimations can be obtained.

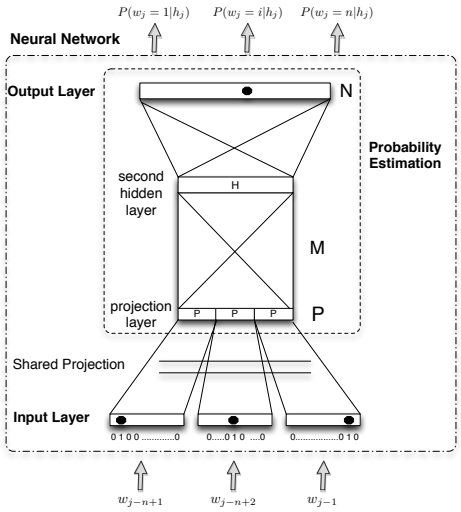


Figure 1: The neural network language model architecture. P, N and H are the size of one projection, one hidden layer and the output layer respectively. h_j denotes the context $w_{j-(n-1)}^{j-1}$.

We report the results using perplexity as well as when these improved CSLMs are integrated into an SMT system. This is performed by re-scoring the n-best list and adding an additional feature function.

2. Modified architecture

The basic architecture of a CSLM with auxiliary data is shown in Figure 2. The example in the figure shows only one additional auxiliary feature vector. This architecture would allow different auxiliary information for each n-gram, but since our goal is to model the topic or long-term context, we made the choice to keep the auxiliary data constant for all n-grams of one sentence. Therefore, the auxiliary data is loaded once for each sentence. If more than one auxiliary feature is desired, the dimension of the auxiliary feature vector will be equal to the sum of the individual feature dimensions. In this case the auxiliary feature vector will be the concatenation of two or more feature vectors. This architecture also allows us to use sentence-level features as well as document (or corpus) level features by using the same auxil-

iary vector for all lines in the document (or corpus).
The functionality of auxiliary features has been integrated in the open-source CSLM toolkit [1] [9].

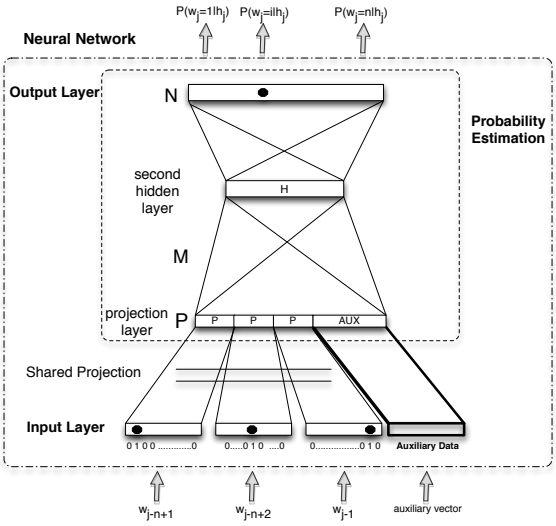


Figure 2: CSLM modified architecture with additional new auxiliary data input to the neural network

3. Related work

Although, we focus on improving CSLM in this work, some related research focus on improving the standard n-gram language models by integrating more context or semantic knowledge. Kuhn and De Mori [12] proposed to calculate the probabilities which correspond to the relative proportion of the last N words. They present a combined LM that interpolates a general trigram LM and another LM called a cache-based LM which is trained on the last N words. The relative interpolation weights assigned to each component are based on the POS of each word. The cache component assigns higher probability to recently encountered words. In our work, the context is represented as a continuous space vector. It can be one line or the whole history back to the beginning of the document. In the latter case more weight is given to recent lines.

Bellegarda [13] proposed a method to use more global constraints to improve LM since local constraints are already captured by the n-gram model. They use latent semantic analysis (LSA) which automatically discovers the semantic relationships between words and documents in a given corpus. In their approach, words and documents are mapped into a continuous semantic vector space, in which clustering techniques are used. This allows the characterization of parallel layers of semantic knowledge in the space, with variable granularity. The resulting LMs complement the conventional n-gram LMs. They suggested to use hybrid n-gram+LSA models to benefit from the advantages of several smoothing techniques.

In a similar work, Coccaro and Jurafsky [14] integrated semantic knowledge into an n-gram LM using LSA and a word similarity algorithm. Since LSA is a bad predictor of frequent words, they used a geometric instead of a linear combination based on a per-word confidence metric. In our work, instead of using LSA , we use the line context vector representations which is calculated using the embeddings of the words in this line. The word embeddings are the projections learned during CSLM training. We were motivated by what was reported recently by Baroni et al. [15] that the context predictive models (i.e. word embedding) outperform classic count-vector-based distributional semantic approaches.

Other works, like the work of Iyer and Ostendorf [16] focused on developing a sentence-level mixture language model that takes advantage of the topic constraints in a sentence or article. They proposed topic-dependent dynamic cache adaptation techniques in the framework of mixture models. An automatic clustering algorithm was used to classify text with two levels of mixture models for smoothing. In our work a predefined genre is assigned to different corpora, which is used as additional input to the neural network. However, it is also possible to use topics instead of genres and to assign the topic dynamically by using similar automatic clustering algorithm like the one used by [16].

Khudanpur and Wu [17] proposed an LM that combines collocational dependencies with the syntactic structure and the topic of the sentence. They integrate these dependencies using a maximum entropy technique. They report a substantial improvement in perplexity and in the accuracy of a speech recognition task. In our work, instead of using topic, we used the genre of the sentence. Since we are using auxiliary features on the sentence level, it could be envisioned to extend our work to use syntactic features.

Mikolov and Zweig [18] focused on improving the performance of recurrent neural network language models (RNNLMs) by using a topic-conditioned RNNLM. They used a contextual real-value input vector in association with each input word. This vector is used to convey contextual information about the sentence being modeled. They use Latent Dirichlet Allocation (LDA) to get a compact vector-space representation of a long span context which they conventionally interpreted as a topic representation. They argue that their approach has the key advantage of avoiding the data fragmentation associated with building multiple topic models on different data subsets. The main differences with our work are, that we used a feed-forward neural network and context vector representation instead of LDA. Also, we evaluated the impact of using various types of auxiliary feature as explained in Section 4.

[1] Available for download from https://github.com/hschwenk/cslm-toolkit

4. Auxiliary features

In this work, we experimented with two types of auxiliary features: the **first** one provides a feature of the current line itself (e.g. the number of words or genre) which allows us to train feature-conditioned continuous space language models. Some of these features are motivated by research in the machine translation quality estimation literature. The **second** type of auxiliary feature aims at providing a larger context. Table 1 summarizes the auxiliary features of this type that we have experimented with. Each auxiliary feature has a reference name that we are using in this paper.

One of the basic auxiliary feature we used is **LineLen** or the line length, expressed in number of words. We used an 1-of-n encoding to generate this feature vector. The ith value in the vector is set to 1 if the line length is equal to i, and zeros otherwise. We considered a maximum line length of $n = 200$, so if the line length exceeded 200 words, we use $n = 200$. In our experiments this 1-of-n encoding is projected into a continuous space like for the words.

Aux feature	Embeddings
CurrLine	words in the **current** line
PrecLine	words in the **preceding** line
PrecHCurrLines	**current** line and **h preceding** lines
AllPrecCurrWords	words in the **current** and **all preceding lines**
AllPrecWords	words in **all preceding** lines
AllPrecLines	**all preceding** lines

Table 1: *Auxiliary features using normalized weighted sum of different embeddings*

The **Genre** consists of a binary vector with dimension equal to the number of genres we have. As for LineLen, we used a 1-of-n encoding. In our training data, we have 5 genres as shown in Table 3.

For the context vector representation auxiliary features, We used various ways to compose them. One of the composition is **CurrLine** $\hat{\alpha}_l$ of a line l. This will be the normalized sum of the word embeddings e_w of all tokens $w \in l$ computed as follows:

$$\hat{\alpha}_l = \frac{\sum_{w \in l} e_w}{||\sum_{w \in l} e_w||} \quad (1)$$

Similarly, **PrecLine** auxiliary feature $\hat{\beta}_l$ is calculated as follows:

$$\hat{\beta}_l = \frac{\sum_{w \in l-1} e_w}{||\sum_{w \in l-1} e_w||} \quad (2)$$

For **PrecHCurrLines**, we calculate the weighted sum of the context vector representation of the current line $\hat{\alpha}_l$ and the preceding H lines. The farther the line is in the past, the lower the weight is. The vector of a line l is calculated as follows:

$$\hat{\eta}_{l,H} = \frac{\sum_{i=l-H}^{l} \hat{\alpha}_i \lambda^{l-i}}{||\sum_{i=l-H}^{l} \hat{\alpha}_i \lambda^{l-i}||} \quad (3)$$

In our experiments we used different values of H=10, 30, 50 and λ=0.95.

The differences between **AllPrecLines** and **PrecHCurrLines** is that the first one does not include the current line context vector representation in the calculation of its vector and that it uses all preceding lines not just the H preceding lines. The equation used to calculate the feature vector of AllPrecLines of a line l is as follows:

$$\hat{\omega}_l = \frac{\sum_{i=1}^{l-1} \hat{\alpha}_i \lambda^{l-i}}{||\sum_{i=1}^{l-1} \hat{\alpha}_i \lambda^{l-i}||} \quad (4)$$

For the first line, we used the context vector representation of itself (i.e. $\hat{\omega}_1 = \hat{\alpha}_1$). In our experiments, we used several weights: $\lambda = 0.85, 0.95, 0.98$.

For **AllPrecCurrWords**, the line context vector representation $\hat{\sigma}_l$ is calculated using all preceding words with a weight λ that gives more weight to the near history **words** and lower weight to the far history **words**. The equation used to calculate the feature vector of AllPrecCurrWords of a line l is the following:

$$\hat{\sigma}_l = \frac{\sum_{i=1}^{W'-1} e_{w_i} \lambda^{W'-i}}{||\sum_{i=1}^{W'-1} e_{w_i} \lambda^{W'-i}||} \quad (5)$$

where W' is the number of words in the current and all preceding lines. In our work we experiment with the following weights: $\lambda = 0.75, 0.85, 0.95$.

AllPrecWords is calculated in a similar way as **AllPrecCurrWords**, but excluding the words of the current line.

5. Evaluation on Penn Treebank

We first evaluated our work on the English Penn Treebank (PTB) corpus [19]. This is a very small corpus ($<$ 1 million words training data), but it has the advantage that many comparable results are published. We limited our evaluation on PTB to use only the preceding line auxiliary feature (i.e. PrecLine). The features **LineLen** and **CurrLine** can not be used when using perplexity to evaluate an LM since they provide information on the future. However, it is valid and useful to apply them in an n-best list re-scoring framework, as discussed later in this paper.

The perplexity values on PTB for several configurations are shown in Table 2. We experiment with different learning rate

scales for the first layer of the neural network as shown in the third column in Table 2. This means that the first layer learning rate is scaled by this value which means that the network learns the weights faster than other layers weights and possibly learns better projection weights. **Copy** means that no weights are learned and the auxiliary feature vector is copied to the next layer directly.
In CSLM1, using auxiliary features and unified learning rate scale decreased the perplexity slightly. The same happen when we replaced **Copy** by a **sequence of double hyperbolic tangent** in CSLM3, and when we increased the learning rate scale to 2 in CSLM4, comparing to Baseline2. Changing the learning rate scale to 3 in CSLM5, again, decreased the perplexity by 7.5 on dev and 7.2 on test vs. Baseline2. So the perplexity of CSLM5 compared to Baseline1 decreased by 7.6% on dev and 7.5% on test.

System	**Aux layer**	**lrs**	**DevSet** PPL	**TestSet** PPL
Baseline1 (No Aux)	-	1	133.19	127.66
Baseline2 (No Aux)	-	2	130.48	125.28
CSLM1	Copy	1	128.26	123.45
CSLM2	Copy	2	124.80	120.32
CSLM3	Seq. of two tanh	1	127.15	121.93
CSLM4	Seq. of two tanh	2	124.22	118.57
CSLM5	Seq. of two tanh	3	**122.98**	**118.08**

Table 2: *Perplexity on Penn Treebank using the PrecLine auxiliary feature with different auxiliary layer topology and learning rate scale (lrs) for the first layer.*

To understand these results, we compared systems with the same setup except for one variable. Comparing Baseline1 and Baseline2 shows the impact of increasing the learning rate scale from unified to 2. Also comparing CSLM1 and CSLM2 gives us the impact related to the increase of learning rate scale for word embeddings only since the **Copy** layer used for auxiliary feature does not have any weights. Also comparing CSLM1 and CSLM3, gives us the impact of using **sequence of double hyperbolic tangent** layer for auxiliary data instead of **Copy**. We observed that this allows the network to deeply learn from the auxiliary data. These three comparisons accumulated a perplexity decrease of 7.28 on dev and 7.03 on test. We concluded that using auxiliary feature decreases the perplexity with different meta-configuration and topology by around 7.5% on dev and test.

6. SMT experimental results

We evaluate the performance of our improved CSLMs which use auxiliary features in the context of SMT. This is done by using them to re-score the n-best list provided by an SMT system. A new CSLM score is added to the n-best list for each hypothesis and the coefficients of all feature functions are optimized. In the following subsections, we describe our baseline system and the rescoring results with some discussions.

6.1. Baseline system of SMS/Chat

The language pair of the baseline system is Arabic Egyptian dialect into the English. The translation task is SMS/Chat translation in the context of DARPA BOLT project. The system is a standard phrase-based system trained using Moses toolkit [21], SRILM [22], KenLM [23], and GIZA++ [20]. Log linear weights are optimized using MERT [20]. We evaluated the translation quality using BLEU [24].
We used the following technique to build our baseline SMT system:

- **Data selection**: We selected the most relevant sentences to the task from the bilingual corpora based on the work of [25] using XenC [26] open source toolkit . The selected sentences are used to train our phrase-based system. Since our SMT system is for SMS/Chat genre, the training data size using data selection is 4.7m words only as shown in Table 3 compared to the full available bilingual corpora size of 191.26m. Another advantage of using data selection is to have smaller translation model. Dev and test sets are shown in Table 4. Dev set is used for tuning the weights of the feature functions.

corpus	corpus genre	selected size Ar/En tokens
smschat	SMS/CHAT (Egyptian)	648k/845k
gale	Modern Standard Arabic (MSA)	128k/158k
e103		44k/46k
fix		73k/84k
ummah		36k/37k
isi		354k/348k
bolt	FORUM (Egyptian)	136k/165k
bbnturk		167k/177k
bbnlev	FORUM (Levantine)	111k/124k
un	FORMAL MSA (UN)	1.34m/1.27m
cts	CALLS (Egyptian)	1.24m/1.45m
Total	-	4.28m/4.7m

Table 3: The size of the selected data from bilingual corpora for SMS/CHAT SMT baseline system

type	**# Arabic tokens**	**# English tokens**	**genre**
dev	19.7k	25.6k	SMS/CHAT
test	19.4k	24.6k	SMS/CHAT

Table 4: *Development and test sets of SMS/Chat SMT system*

- **Data weighting**: This method is used to weight the bilingual sub-corpora models according to their importance to the translation task. We used a method based on the work of [27] using perplexity minimization given the development set. if $\overline{s}$ and $\overline{t}$ denote the source and target phrase respectively, we are instantly optimizing the weight of the four features: $p(\overline{s}|\overline{t})$, $lex(\overline{s}|\overline{t})$, $p(\overline{t}|\overline{s})$ and $lex(\overline{t}|\overline{s})$ in the Moses translation model.

- **Language modeling**: We used data selection method based on [28] to select the relevant monolingual data for our 4-gram back-off language model. The back-off LM was used in SMT decoding for generating the 1000-best translation output. We used this back-off LM also in CSLM re-scoring to calculate the probability of words not in the CSLM shortlist.

type	data set	# English tokens	genre
train	gale	5.01	MSA
	bolt	2.05m	FORUM (Egyptian)
	smschat	845k	SMS/CHAT
	Total	7.9m	-
dev	smschat dev	25.6k	SMS/CHAT

Table 5: *Training corpora and dev set used to train and tune the CSLM models*

6.2. Result and analysis of re-scoring the n-best list

CSLM models with various auxiliary features were trained using CSLM toolkit on three English corpora (total of 7.91m words) which are the target side of the bilingual corpora shown in Table 5.
The results obtained by re-scoring the n-best list created by the baseline system are summarized in Table 6. The table contains the best result for each auxiliary feature. Detailed results can be found in Tables 7 and 8. Since the test set BLEU scores of both **SMT Baseline** and **CSLM Baseline** without auxiliary data are the same, we decided to use SMT Baseline as the Baseline for the result analysis.

The CSLMs English training corpora used in these experiments is about 7.9m tokens (see Table 5). These results were obtained with the best meta-parameters (i.e. H and λ). In Table 6, we described the CSLM model, auxiliary feature dimension, auxiliary feature projection dimension along with the BLEU scores on dev and test. We used **projection** layer for **LineLen** auxiliary feature, **Copy** layer for **Genre** auxiliary feature, **sequence of double hyperbolic tangent** layer for the rest of auxiliary features. All experiments are trained with 24-gram context size.

System	Aux dim/proj.	Dev	Test
SMT Baseline	-	27.35	25.72
CSLM Baseline (No AuxData)	-	28.04	25.67
LineLen	1/200	28.65	26.14
Genre	5/-	**28.90**	**26.32**
CurrLine	320/-	28.29	26.09
PrecLine	320/-	28.67	**26.33**
PrecHCurrLines λ=0.95, h=50	320/-	**28.92**	26.26
AllPrecCurrWords λ=0.75	320/-	28.52	25.86
AllPrecWords λ=0.95	320/-	**28.77**	**26.82**
AllPrecLines λ=0.98	320/-	28.63	**26.52**

Table 6: BLEU *scores obtained when re-scoring the n-best list using different auxiliary data.*

Looking at Table 6, we observed a good improvement using *LineLen* auxiliary feature, but *Genre* has relatively better gain on both dev and test. This means that *Genre* is better discriminative auxiliary feature.

We observed that *PrecLine* provides better performance due to better context information compared to *CurrLine*. We also observed that CSLMs with auxiliary features which contain the current line (i.e. *AllPrecCurrWords*, *PrecHCurrLines*) generally have lower BLEU scores than CSLMs with auxiliary features which do not contain the current line. We concluded that using current line is not so useful for re-scoring n-best list because instead of predicting the next word, the CSLM would rather learn to find the next word from the input auxiliary feature making undesirable cycle in the model.

PrecLine has +0.6 BLEU gain on test. If one preceding line is useful, two or more preceding lines would be more useful (possibly weighted). We can verify this assumption by looking at *AllPrecLines* result, which uses auxiliary feature that does not contain the current line (i.e. both *AllPrecCurrWords*, *PrecHCurrLines* contain the current line). The results of *AllPrecLines* is 26.52 on test which is the second best BLEU score in Table 5, which confirms that our assumption is correct.

Looking at the additional results of *AllPrecLines* with different λ(s) in Table 7, we observed that larger λ weight improved the BLEU score on both dev and test sets. The best BLEU scores are obtained using *AllPrecWords* CSLM. The only difference between *AllPrecLines* and *AllPrecWords* is that the second one is weighted sum of words' embeddings, while the first one is the weighted sum of lines' embeddings.

System	λ	Dev	Test
SMT baseline	-	27.35	25.72
CSLM Baseline	-	28.04	25.67
CurrLine	-	28.29	26.09
PrecLine	-	28.67	26.33
AllPrecLines	0.85	28.06	25.52
AllPrecLines	0.95	28.59	26.42
AllPrecLines	0.98	**28.63**	**26.52**
AllPrecWords	0.75	28.37	26.36
AllPrecWords	0.85	28.74	26.49
AllPrecWords	0.95	**28.77**	**26.82**
AllPrecCurrWords	0.75	**28.52**	**25.86**
AllPrecCurrWords	0.85	28.23	25.59
AllPrecCurrWords	0.95	28.21	25.64

Table 7: BLEU *scores of re-scoring n-best list using AllPrecLines, AllPrecWords and AllPrecCurrWords auxiliary features with various weights. Auxiliary layer is a sequence of two tanh 320x320.*

It means that *AllPrecWords* auxiliary feature includes better and consistent context information. One possible reason for this is that for *AllPrecLines* auxiliary feature vector, each line has a different length, and hence the *weight* on each line controls the contribution of a variable number of words. This clearly is less stable than using the weighted sum of individual words embeddings and hence the auxiliary feature vector will be independent of individual lines lengths. In Table 7, we noticed the same relation between λ and the BLEU scores as we discussed for *AllPrecWords* auxiliary feature.
Looking at the results of *AllPrecCurrWords* auxiliary feature in Table 7, we observed that the results also are inconsistent on test, λ=0.75 gives better scores than λ=0.85, but also, λ=0.95 gives better scores than λ=0.85. We concluded that including word embeddings of both current line and preceding lines in the same auxiliary feature gives inconsistent results. For the results of *PrecHCurrLines* in Table 8, generally, we observed that including more preceding lines does not give better scores on test (we used maximum 50 preceding lines in these experiments), even with H=50, the scores are not better than just one preceding line **PrecLine**. We concluded that the reason is that this auxiliary feature includes the current line embeddings which cause inconsistent results on dev and almost no improvement on test.

System	H	Dev	Test
SMT baseline	-	27.35	25.72
CSLM Baseline	-	28.04	25.67
CurrLine	-	28.29	26.09
PrecLine	-	28.67	**26.33**
PrecHCurrLines	10	28.70	26.21
PrecHCurrLines	30	28.28	**26.26**
PrecHCurrLines	50	**28.92**	**26.26**

Table 8: BLEU *scores using PrecHCurrLines auxiliary feature with number of preceding lines H and* $\lambda = 0.95$. *Auxiliary layer is a sequence of two tanh 320x320.*

7. Conclusions

In this paper we introduced a novel method to improve the continuous space language model using auxiliary features. We used different features which some of them are motivated by the important features in machine translation quality estimation literature. The suggested auxiliary features include text genre, line length and various types of context vector representations.

We reported perplexity improvement around 7.5% on dev and test using the English Penn Treebank dataset. We also reported an improvement on a translation task up to 1.42 BLEU on dev and 1.1 on test by re-scoring n-best list of a strong baseline phrase-based SMT system. Also, the results show that the weighted sum of the word embeddings is more stable and outperforms the line level weighted sum of embeddings. These results need to be validated on other tasks with different language pairs, genres and data sets.

In future work, we would like to try using combined features and explore syntactic features. Also we would like to experiment with additional features like source language features and study their impact on the CSLM performance.

8. Acknowledgements

This research was partially financed by DARPA under the BOLT contract.
We would like to thank the reviewers of this paper for their helpful comments.

9. References

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944966

[2] H. Schwenk, "Efficient training of large neural networks for language modeling," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 4. IEEE, 2004, pp. 3059–3064.

[3] H. Schwenk, H. Schwenk, and J.-l. Gauvain, "Connectionist language modeling for large vocabulary continuous speech recognition," *IN INTERNATIONAL CON-*

FERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, pp. 765–768, 2002.

[4] H. Schwenk, "Continuous Space Language Models," vol. 21, no. 3, pp. 492–518, 2007.

[5] J. Park, X. Liu, M. J. Gales, and P. C. Woodland, "Improved neural network based language modelling and adaptation." in *INTERSPEECH*, 2010, pp. 1041–1044.

[6] L. Lamel, J.-L. Gauvain, V. B. Le, I. Oparin, and S. Meng, "Improved models for mandarin speech-to-text transcription," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4660–4663.

[7] H. Schwenk, D. Déchelotte, and J.-L. Gauvain, "Continuous space language models for statistical machine translation," in *Proceedings of the COLING/ACL Conference*. Morristown, NJ, USA: Association for Computational Linguistics, 2006, pp. 723–730.

[8] H. Schwenk, "Investigations on large- scale lightly-supervised training for statistical machine translation," in *IWSLT*, 2008, pp. 182–189.

[9] ——, "Continuous space language models for statistical machine translation," in *The Prague Bulletin of Mathematical Linguistics, (93):137–146.*, 2010.

[10] H. S. Le, I. Oparin, A. Messaoudi, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Large vocabulary soul neural network language models." in *INTERSPEECH*, 2011, pp. 1469–1472.

[11] H. Schwenk, "Continuous space translation models for phrase-based statistical machine translation," in *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Posters, 8-15 December 2012, Mumbai, India*, M. Kay and C. Boitet, Eds. Indian Institute of Technology Bombay, 2012, pp. 1071–1080. [Online]. Available: http://aclweb.org/anthology/C/C12/C12-2104.pdf

[12] R. Kuhn and R. De Mori, "A cache-based natural language model for speech recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 6, pp. 570–583, 1990.

[13] J. R. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1279–1296, 2000.

[14] N. Coccaro and D. Jurafsky, "Towards better integration of semantic predictors in statistical language modeling." in *ICSLP*. Citeseer, 1998.

[15] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2014, pp. 238–247.

[16] R. M. Iyer and M. Ostendorf, "Modeling long distance dependence in language: Topic mixtures versus dynamic cache models," *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 30–39, 1999.

[17] S. Khudanpur and J. Wu, "Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling," *Computer Speech & Language*, vol. 14, no. 4, pp. 355–372, 2000.

[18] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *2012 IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA, December 2-5, 2012*. IEEE, 2012, pp. 234–239. [Online]. Available: http://dx.doi.org/10.1109/SLT.2012.6424228

[19] 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.

[20] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Comput. Linguist.*, vol. 29, pp. 19–51, March 2003.

[21] 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 *Meeting of the Association for Computational Linguistics*, 2007, pp. 177–180.

[22] A. Stolcke, "Srilm - an extensible language modeling toolkit," *In Proceesings of the 7th International Conference on Spoken Language Processing (ICSLP 2002*, pp. 901–904, 2002.

[23] 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. [Online]. Available: http://kheafield.com/professional/avenue/kenlm.pdf

[24] 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*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318.

[25] 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.: Association for Computational Linguistics, July 2011, pp. 355–362.

[26] 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.

[27] 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, April 2012, pp. 539–549. [Online]. Available: http://www.aclweb.org/anthology/E12-1055

[28] 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. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858842.1858883

Multifeature Modular Deep Neural Network Acoustic Models

Kevin Kilgour, Alex Waibel

International Center for Advanced Communication Technologies - InterACT,
Institute of Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
kevin.kilgour,alexander.waibel@kit.edu

Abstract

This paper presents and examines multifeature modular deep neural network acoustic models. The proposed setup uses well trained bottleneck networks to extract features from multiple combinations of input features and combines them using a classification deep neural network (DNN).

The effectiveness of each feature combination is evaluated empirically on multiple test sets for both a classical DNN as well as a for modular DNNs using only a single module. Modular DNNs using two or more modules are shown to reduce the WER by up to 11.5% relatively compared to a baseline DNN and give the best overall performance on both test sets.

1. Introduction

The first step in speech recognition is to extract a stream of feature vectors form the audio. Although many of these so called front-ends are fundamentally similar and equally useful, they are still, to some extent, complementary and the outputs of ASR systems trained separately on different front-ends can be combined in such a manner that the combined output contains fewer transcription errors than either of the individual outputs [1, 2]. While very useful, this high level combination method has the disadvantage of requiring multiple ASR systems to be run in parallel.

In this paper an alternative approach is proposed that uses modular deep neural networks (mDNNs [3]) to combine the features in a single acoustic model. An mDNN can be seen as an extension of a time-delay neural network (TDNN) [4]. TDNNs are designed to be time invariant and work on sequences of feature vectors. As well as the current feature vector x_t the neurons of the first hidden layer are also connected to a few of the preceding feature vectors $x_{t-1}x_{t-2}, ...$ The *time-delay* procedure is applied at the transition from the first hidden layer to the second hidden layer. The neurons of the second hidden layer also possess connections to the first hidden layer's outputs at the preceding steps.

While both TDNNs [5] and CNNs [6, 7] may have many *time-delayed* or convolutional layers these layers are normally directly connected to their preceding layers. Modular DNNs on the other hand use well trained deep neural networks to connect the input layer to the *time-delayed* layer. As these network modules are trained as bottleneck features (BNF) [8] we refer to this layer as the bottleneck layer. In this paper we show how using multiple features as inputs to the BNF modules can improve the performance of an mDNN and go on to experiment with using an mDNN to combine multiple different BNF modules.

This paper is structured as follows: after an overview of the relevant related work in section 2 a multifeature DNN AM is introduced as well as all the features used throughout this paper. This is followed in section 3 by a description of the proposed multifeature DNN. Section 5 explains how the neural networks are evaluated the presents and results after which section 6 concludes the paper with a short summary.

2. Related Work

A method of using BNFs to combine multiple feature streams proposed in [9], shows that combining MFCC, PLP and gammatone features in the input layer of an MLP can lead to a system that performs better than the system combination of the lattices of the individual systems. The MLP using the combined input feature also outperforms the best single feature MLP by a small amount. In contrast to this work they, however, only look at shallow networks. Instead the authers later focus on integrating the multiple features into a shallow RNN [10] trained to classify the phone targets. Stacking MFCCs and MVDRs (Minimum Variance Distortionless Response) at the input of a DNN was also found to be helpful in bottleneck feature extraction for German Broadcast News [11] as well as for the NIST 2013 OpenKWS evaluation [12].

In [13] various tonal models and methods of integrating tonal features are analyzed on both tonal and non tonal languages. That work reports, for the first time, results of using fundamental frequency variation [14] features for speech recognition of tonal languages and finds that early integrating tonal features consistently leads to a reduction in WER even on non-tonal languages.

A version of the modular DNN designed for low resource languages is discussed in [3] and modified to make use of language resources outside of the target language [15].

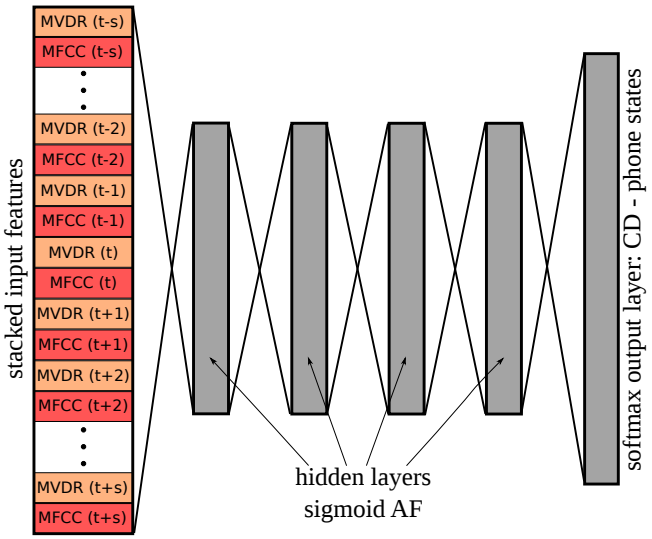


Figure 1: *An example of a deep neural network with 4 hidden layers and an output layer with corresponding CD phone states. Its input is a 2s+1 frame window containing both MFCC and MVDR features.*

3. Multifeature Deep Neural Network Accoustic Models

Multifeature DNN-AMs are feed forward neural networks that merge multiple different input feature in the input layer of the neural network. An example DNN AM is shown in figure 1. It has an input layer that uses stacked MVDR+MFCC features in a window spanning from s frames prior to the current frame to s frames after the current time frame, followed by four hidden layers that use the sigmoid activation function and a softmax output layer where the neurons correspond to the context dependent phone states. The example contains two of the four different input feature used in this paper:

- **Mel Frequency Cepstral Coefficients (MFCC)**: MFCCs have established themselves as the most common front-end feature in speech recognition. They are computed by applying a discrete cosine transformation to log Mel features.
- **Minimum Variance Distortionless Response (MVDR) Spectrum**: MVDR [16] features are an improvment on basic linear prediction features [17]. In some circumstances warped Minimum Variance Distortionless Response (MVDR) features for speech recognition have been shown to be better than MFCC features.

The other two features used in this paper are:

- **Log-MEL Features (lMEL)**: Motivated by the physiology of human hearing, the mel scale developed by [18] is applied after performing a short-time Fourier transformation of the audio. In large DNN AMs lMEL features tend to outperform MFCC.

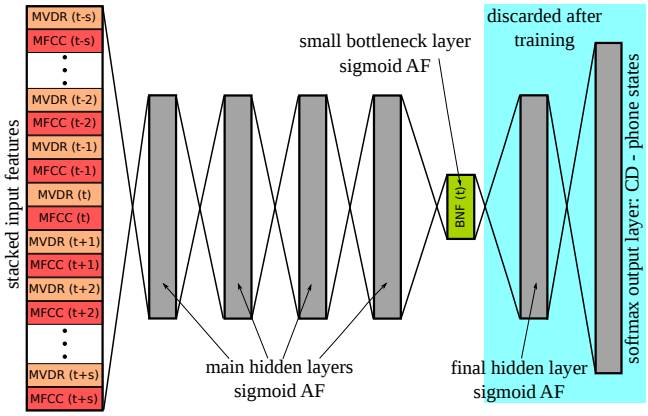


Figure 2: *An example DBNF with 4 hidden layers prior to the bottleneck layer*

- **Tonal Features**: The tonal features used are a combination of Kjell Schubert's [19] pitch based features and Laskowski's [14] FFV features. Tonal features are not suitable as stand-alone features and are only used to augment the other features.

4. Multifeature Modular Deep Neural Network Accoustic Models (mDNN-AMs)

In this section the proposed modular deep neural network acoustic model is introduced. Feed forward DNN AM such as the one depicted in figure 1 have been analyzed and examined by many authors [20, 21]. Both their observations and ours indicate that the addition of further hidden layers does not result in any noticeable improvement for DNNs after about 5-8 layers. One possible explanation is that the lower layers are being poorly trained because the gradient decreases with each layer it is passed back though. In order to solve this problem the mDNN-AMs use a well trained bottleneck feature (BNF) module as the basis for the DNN-AM.

4.1. Multifeature Bottleneck Feature (BNF) Modules

In an MLP the outputs of a given layer can be thought of as an alternative representation of the input feature vector. A small hidden layer in the middle of an MLP will, therefore, provide compact alternative features, with the number of coefficients being controlled by the number of neurons in the hidden layer.

An example of such a network with 4 hidden layers prior to the bottleneck layer is given in Figure 2. Bottleneck features are discriminatingly trained using context dependent subphone states as targets. After training all the layers following the bottleneck are discarded. In a classic GMM system BNFs are typically used to transform the input feature stream into a stream of bottleneck features which are then stacked over a temporal window. An LDA or PCA is then used to reduce the dimension back to the desired input size for the GMM.

Multifeature-BNFs concatenate multiple different features into a single large input vector. Using combinations of the features described in section 3 seven different BNF modules are analyzed in this paper.

4.2. mDNN Topology

An example modular DNN (mDNN) AM using MVDR+MFCC features is shown in figure 3. Two features (MFCC & MVDR) are extracted at each frame and used together with their neighbours in a stack as the inputs to a BNF network.

The final layers of a modular DNN-AM are same as the final layers in a normal DNN-AM. Instead of a normal input layer the modular DNN-AM has a bottleneck layer which consists of stacked BNF frames from an already fine-tuned BNF network. We refer to those final layers as the classification module (or DNN-module) and after integration the BNF network is referred to as the BNF module. If the classification module has an input context of $2r + 1$ (r-BNF frames before and after the current frame) and the BNF module has a input context of $2s + 1$ then the total network requires an input context of $2(r + s) + 1$ frames. For the BNF frame at $t - i$ the input frames from $t - i - s$ to $t - i + s$ have to be stacked and used as the input to the BNF-module. The BNF-module is applied $2r + 1$ times to generate each of the $2r + 1$ BNF frames in the BNF layer.

4.3. Weight Tying

During fine tuning the weights of the BNF-module are tied. Errors can be propagated back past the BNF-layer into all applications. Weight tying allows the modular DNN-AM to continue on using a single BNF-module. Its weights are updated using the average update:

$$\Delta w_j = \sum_{k=1}^{2r+1} \Delta w_j^k \tag{1}$$

such that a single BNF-module learns to produce BNFs that can be used in any part of a stack BNF layer.

4.4. Integration

Although the total computation cost for a single frame is very high, the BNF frames can be cached and reused for the next frame. At frame t the DNN-module of the example mDNN in figure 3 requires the output of the BNF-module for $2r + 1$ different inputs from $t-r$ to $t+r$. For frame $t-1$ it requires outputs from the BNF-module for the inputs from $t - 1 - r$ to $t - 1 + r$. So with the exception of the output of the BNF-module at $t + r$ all of the required outputs for frame t have already been produced and cached.

The BNF-module is simply used to convert a stream (or multiple streams) of input features into a stream of BNF features which are then used as the input stream for the DNN-module. In an offline setting the stream of input feature vectors from an utterance forms a matrix and the BNF-module converts this matrix into a matrix where the columns are BNFs.

Because it is faster to perform a single matrix times matrix operation using a fast BLAS (Basic Linear Algebra Subprograms) library than it is to perform many vector times matrix operations, it makes sense to compute the first hidden layer for all features at the same time. So in an mDNN, if T features are extracted from the audio of an utterance they are first transformed into T activations of the first hidden layer of the BNF-module, then to T activations of the 2nd, 3rd and so on hidden layers and then to T bottleneck features followed by T activations of the first hidden layer of the DNN-module. After transitioning though all the hidden layers the T probability distributions over the cd-phone states are all produced at the same time.

If T features are extracted from the utterance then computing the BNF at frames $t = T$ or $t = 1$ could be problematic since these frames require information about the features at $t = T + r$ or $t = -r$. To solve this every requested feature vector prior to the first one is set to the value of the first feature and the final feature vector is used for every feature that could follow it. The same out of bounds rule is applied when the BNFs are used as an input to the DNN-module of the mDNN.

4.5. Modular DNNs with multiple BNF-modules

The modular DNN is not restricted to a single BNF-module and can use multiple BNF-modules at the same time. Figure 4 shows an example mDNN with two 4 layer BNF-modules. The upper BNF-module uses MFCC features as its input and the lower BNF-module uses MVDR features as its input. Although both networks in this example have an input window of $2s + 1$ frames they could, if it were desired, have different sized input windows and they could also have a differing number of hidden layers. The outputs of both module's BNF layers are concatenated and stacked over a $2r + 1$ window.

5. Experimental Evaluation

All experiments are performed on both the German 2010 Quaero evaluation set (eval2012 [22]) which contains 3 hours and 34 minutes of broadcast news and conversational speech as well as on the 2 hour German IWSLT 2012 development set (dev2012 [23]) that contains TED talks. The results of the systems are measured using WER and reported with an accuracy of two decimal places for the larger eval2012 test set and with an accuracy of a single decimal place for smaller dev2012 test set on which differences smaller than 0.1% would not be statically significant. Statical significance is measured using McNemer's test of significance.

5.1. Speech Recognition System

The decoding and GMM AM training uses the *Janus Recognition Tool-kit* (JRTk) with the Ibis single pass decoder [24].

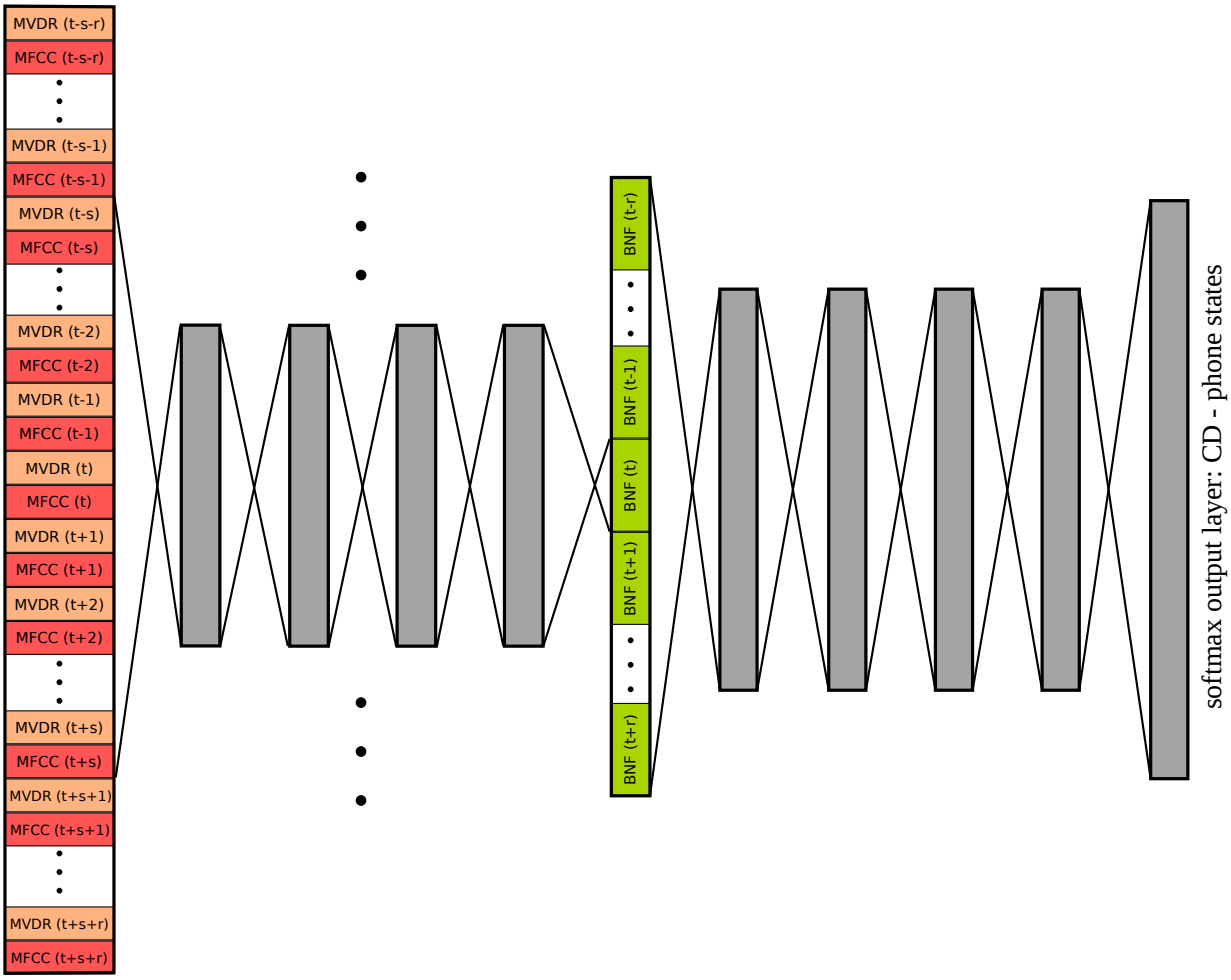


Figure 3: *An example mDNN with 4 hidden layers in the BNF-module and 4 hidden layers in the DNN-module. The input of the DNN-module requires* $2r+1$ *outputs of the BNF-module at different frames. The BNF-module uses a* $2s+1$ *frame window as its input so the whole mDNN network requires* $2(s+r)+1$ *input features which in the example are a concatenation of both MVDR and MFCC features.*

The JrTk [24] is extended with a DNN AM object that implements the same interface to the Ibis decoder as the existing GMM AM. A diverse range of topologies are supported by allowing their computation to be controlled by a tcl script. All acoustic models use a left to right HMM without skip states where each of the 46 normal phonemes have three HMM states and the silence phoneme has only a single state. The cluster tree is built with 6016 leaves.

5.2. Neural Network Training

Each neural network is pretrained layerwise using denoising autoencoders with a 20% corruption and a constant learning rate for 2 million mini batches. After pretraining the final layer is added, with the output layer using the softmax activation function. The full DNN is then fine-tuned using the newbob learning rate schedule. All training is performed using Theano[25].

5.3. Analysis of Multifeature DNNs

Since tonal features can only be used as augmented features and not as individual feature the following combination of input feature were tested:

- Single feature: MFCC, MVDR, lMEL
- 2 features: MFCC+MVDR, lMEL+T[1]
- 3 features: MFCC+MVDR+T
- 4 features: MFCC+MVDR+lMEL+T

Both the MVDR and MFCC features use 20 coefficients while the lMEL features have 40 coefficients and are the same size as the merged MVDR+MFCC feature vector. The addition of 14 tonal features brings the input sizes up to 54 for both the MVDR+MFCC+T (*m2+t*) and the lMEL+T

[1] T=tonal

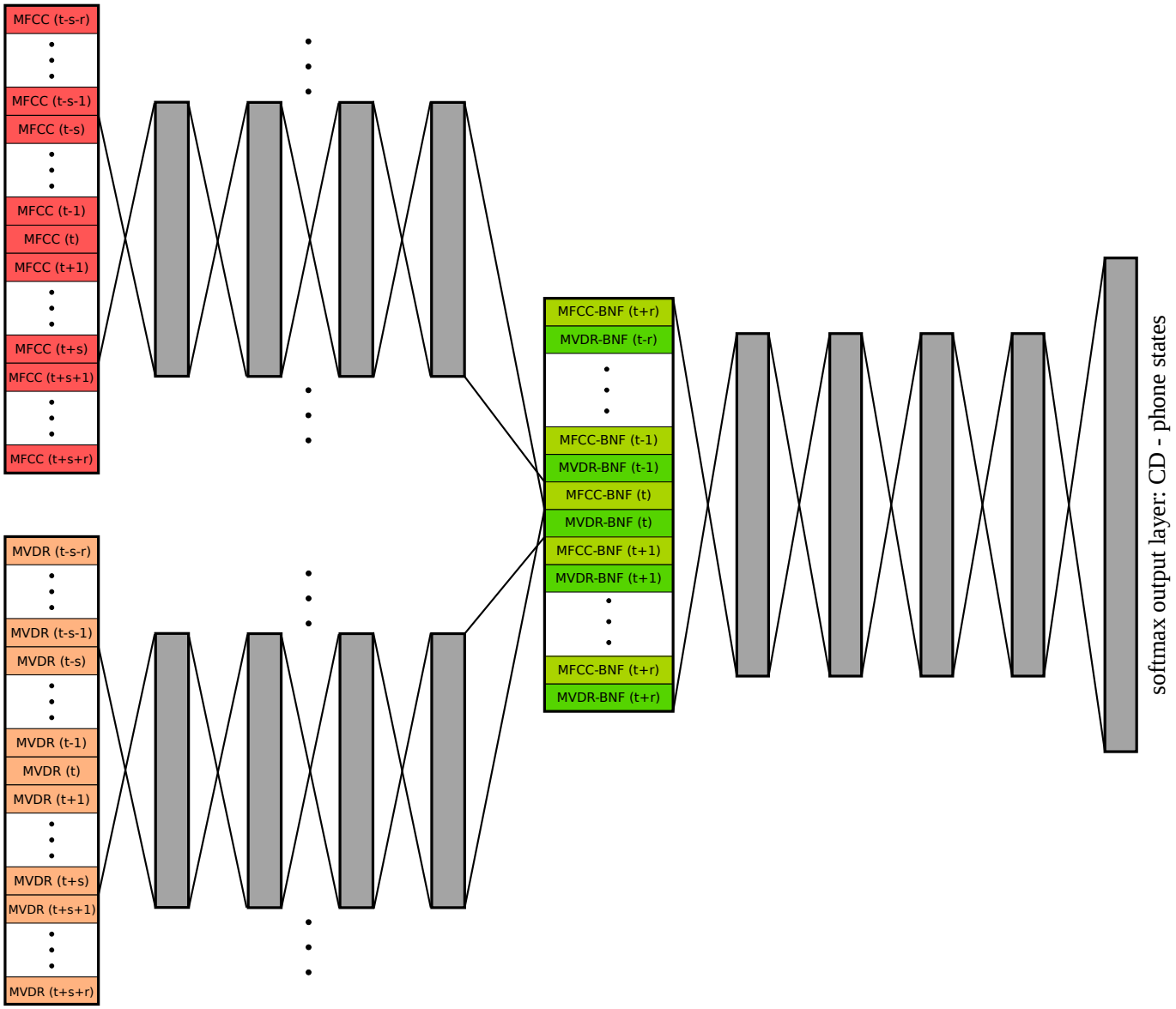


Figure 4: *An example mDNN with a 4 layer DNN-module built on top of two BNF-modules: a 4 layer MFCC BNF-module and a 4 layer MVDR BNF-module. The input of the DNN-module requires* $2r + 1$ *outputs of both the BNF-modules at different frames.*

(*lmel+t*) networks. The final MVDR+MFCC+T+lMEL (*m3+t*) MLP that contains all available features has an input of 94.

Each feature combination is trained on multiple topologies that only vary in their size (1200-2000) and number (4-8) of hidden layers. The reported numbers always use the best topology for the feature combination.

Combining the MVDR and MFCC input features results in a network with a significantly ($p < 0.005$) lower WER than either of the individual features and on about par with networks using the lMEL feature that have the same number of coefficients. The lMEL system is 0.06% better on the eval2010 test set but 0.2% poorer on the dev2012 test set. A pattern in the results shown in table 1 can be found that suggests that input features using more coefficients tend to perform better. The addition of tonal features always results in an improvement.

The best results on the eval2010 test set are achieved by using all input features which is slightly but significantly ($p < 0.005$) better than the *lmel+t* DNNs.

5.4. Evaluation of Multifeature mDNNs

The mDNN topology is evaluated using the same combinations of input features used in the previous experiment and compared with those results. The BNF-modules used in the experiment are taken from working GMM systems where the use of the multifeature deep BNFs were evaluated.

	BNF-Module Name	Features	Best normal DNN		Modular DNN		Comparable CNC	
			eval2010	dev2012	eval2010	dev2012	eval2010	dev2012
MFCC	mfcc	1	15.88	20.3	15.35	19.5	-	-
+MVDR	m2	2	15.45	19.9	14.71	19.4	15.55	20.0
+T	m2+t	3	14.96	19.8	14.54	19.3	-	-
+lMEL	m3+t	4	14.72	19.4	**14.31**	**18.9**	15.09	19.6
lMEL	lmel	1	15.31	20.1	14.72	19.5	-	-
+T	lmel+t	2	14.77	19.6	14.52	19.0	-	-
MVDR	mvdr	1	15.58	20.2	14.81	19.5	-	-

Table 2: *Results of the multifeature mDNNs compared with both normal DNN using the same input feature combinations and equivalent confusion network combinations. Tested on both the eval2010 and dev2012 test set.*

	eval2010	dev2012
MFCC	15.88	20.3
+MVDR	15.45	19.9
+T	14.96	19.8
+lMEL	**14.72**	**19.4**
lMEL	15.31	20.1
+T	14.77	19.6
MVDR	15.56	20.2

Table 1: *Evaluation of DNNs using various combinations of MFCC, MVDR, T and lMEL input features. Result presented on the IWSLT dev2012 and Quaero eval2010 test sets.*

The DNN-modules are pretrained by first mapping the input features into the bottleneck feature space and performed by training and stacking denoising autoencoders. After pretraining the classification layer is added and the whole mDNN network is jointly finetuned. The input feature sizes range from 20 for the mfcc network features to 94 for the m3+t network. With r, the number of BNF frames before and after the current frame used as the input to the DNN-module, set to 7 and each BNF layer containing 42 neurons the DNN-modules input layer has 630 neuron. All mDNN networks have the same topology. Both their BNF-modules and their DNN-modules have 4 hidden layers of 2000 neurons. The whole network, therefore, has 9 hidden layers.

The results of this experiment are shown in table 2. As a comparison, for each input feature combination its best result with a normal DNN, regardless of the topology, is shown in columns of the table. The last column contains a comparison to a system combination using confusion networks performed on the DNN output lattices of single feature DNNs. The cnc comparison result in line two of table is a combination of the best MVDR DNN and the best MFCC DNN and although it is slightly better than both of them it is outperformed by both the MVDR+MFCC DNN and the MVDR+MFCC mDNN. The cnc comparable to the m3+t network is a combination of the MFCC DNN, the MVDR DNN, and the *lmel+t* DNN and does not even improve on the performance of the lmel+t DNN.

For all input feature combinations the mDNN outperforms the normal DNN by 0.5% absolute or more on the dev2012 test set. On the eval2010 test set the improvements varied from an improvement of 0.25% on the lmel+t features to over 0.7% on both the MVDR and MVDR+MFCC features. The relative usefulness of features is not altered by using an mDNN. With 19.4% on dev2012 the *m3+t* DNN has 4.5% relative lower WER than the basic MFCC DNN which has a WER of 20.3 and a 3.5% lower WER than the lMEL DNN which is the best single feature DNN. In the modular case the improvements are slightly less. All single feature mDNNs have a WER on dev2012 of 19.5% and the *m3+t* network has a 3% lower WER at 18.9%. For the single feature inputs the mDNN results in improvements of 3-4% compared to the normal DNN while the multifeature inputs are only improved by 2.5-3.5%.

Using only lMEL features as inputs performs as well as using the combined MVDR+MFCC feature in both the DNN and the mDNN and on both test sets. The addition of tonal features boosts the performance of the lMEL DNN and mDNN more than the MVDR+MFCC DNN and mDNN.

In total the best multifeature mDNN reduces the WER of a basic MFCC DNN by 7% relative from 20.3% to 18.9% on the dev2012 test set and by 10% relative from 15.88% to 14.31% on the eval2010 test set. Compared to the best single feature DNN, lMEL, it still improves the dev2012 test set by 6% and the eval2010 test set by 6.5%.

5.5. Evaluation of mDNNs with multiple BNF-modules

The effectiveness of the mDNN with multiple BNF-modules is evaluated by training 8 mDNNs with between two and seven BNF-modules. The results are compared to performing a CNC on normal DNN networks that use the same input features and the BNF-modules. The BNF-modules are the same as in the multifeature mDNN experiment and can themselves contain multiple input features. After mapping the training data into the bottleneck feature spaces of all BNF-modules used. The DNN-module is pretrained on the merged BNF features. All other training parameters are the same as in the previous experiments.

In all cases the mDNN outperformed the confusion network combination of DNN systems using the same input features. The best mDNN with multiple BNF-modules m2+t $\oplus$ lmel+t $\oplus$ m3+t ($\oplus$ is used to indicate that a combination of

	BNF-Modules	Features	Name	Modular DNN		Comparable CNC	
				eval2010	dev2012	eval2010	dev2012
mfcc	1	1		15.35	19.5	-	-
⊕ mvdr	2	2	sys01	14.54	19.2	15.55	20.0
⊕ m2	3	2	sys03	14.73	19.3	15.44	19.9
mfcc ⊕ mvdr ⊕ lmel+t	3	4	sys02	14.24	18.7	15.09	19.6
m2+t ⊕ lmel+t	2	4	sys04	14.19	18.8	14.68	19.4
⊕ m3+t	3	4	sys08	14.06	18.7	14.45	19.2
⊕ mfcc ⊕ mvdr	5	4	sys06	14.33	18.9	14.83	19.4
⊕ m2 ⊕ lmel	7	4	sys05	14.44	18.8	14.69	19.4
m2 ⊕ lmel	2	4	sys07	14.34	19.1	15.07	19.8

Table 3: *Comparison of mDNNs using multiple BNF-modules with confusion network combinations of normal DNNs using the same input features. The ⊕ is used to indicate that multiple BNF-modules are combined in a single mDNN.*

BNF-modules) improves the best single module mDNN by 0.2% from a WER 18.9% to 18.7% on the dev2012 test set and by 0.25% from 4.31% to 14.06% on the eval2010 test set. Using McNemar's significance test this is found to be significant at $p < 0.005$. The overview of the results given in table 3 begins with a single BNF-module mDNN using mfcc input features that achieves a WER of 15.35% on eval2010 and 19.5% on dev2012. The next entry augments that mDNN with an MVDR BNF-module and improves dev2012 by 0.3% to 19.2% and eval by 0.81% from 15.35% to 14.54%. The further addition of the MVDR+MFCC BNF-module degraded the dev2012 test set to 19.3% and the eval2010 test set to 14.73%. If instead of the MVDR+MFCC BNF-module the lmel+t BNF-module is added to the mfcc ⊕ mvdr mDNN then it is further improved to 14.24% on eval2010 and 18.7% on dev2012.

The best mDNN with two BNF-modules is the m2+t ⊕ lmel+t mDNN which has a WER of 14.19% on eval2010 and 18.8% on dev2012. The addition of an m3+t BNF-module improves it slightly by 0.13% to 14.06% on the eval2010 test set and by 0.1% to 18.7% on the dev2012. The further inclusion of both the MVDR BNF-module and the MFCC BNF-module slightly increases the WER on both sets. Increasing the number of BNF-modules to 7 by also including the m2 and lmel BNF-modules into the mDNN results in another slight increase in WER.

The usefulness of tonal features can be clearly seen by comparing the m2 ⊕ lmel mDNN to the m2+t ⊕ lmel+t DNN which add tonal features to the input to both of the BNF-modules. They are able to improve the dev2012 test set by 0.3% and the eval2010 test set by 0.15%.

Using an mDNN with multiple BNF-modules increases the mDNNs overall improvement compared to an MFCC-DNN by 8% relative on the dev2012 test and by 11.5% on the eval2010 test set. Compared to an lMEL DNN it reduced the WER by 7% relative from 20.1% to 18.7% on dev2012 and by 8% relative from 15.31% to 14.07%.

6. Conclusion

The modular deep neural network acoustic model presented in this paper incorporates the well trained feature extraction networks using multiple input features. It is initially evaluated using only a single feature extraction module. This evaluation demonstrates the usefulness of using multiple different input feature vectors. Modular deep neural networks, whose sole feature extraction network uses multiple features, outperform those using fewer features or even a single feature.

Using two or more different feature extraction networks as modules in the same modular deep neural network results in further improvements. The best approaches use three feature extraction networks that are, in turn, each trained using multiple input features. The best modular deep neural network is able to reduce the word error rate on the test data sets by up to 11.5% relative improvement compared to a baseline deep neural network..

7. Acknowledgements

The authers would like to thank the reviewers for their constructive comments.

The work leading to these results has received funding from the European Union under grant agreement $n \circ 287658$.

8. References

[1] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[2] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 347–354.

[3] J. Gehring, W. Lee, K. Kilgour, I. R. Lane, Y. Miao, A. Waibel, and S. V. Campus, "Modular combination

of deep neural networks for acoustic modeling." in *INTERSPEECH*, 2013, pp. 94–98.

[4] 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, 1989.

[5] H. SAWAI, Y. MINAMI, M. MIYATAKE, A. WAIBEL, and K. SHIKANO, "Connectionist approaches to large vocabulary continuous speech recognition," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 74, no. 7, pp. 1834–1844, 1991.

[6] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, p. 310, 1995.

[7] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8614–8618.

[8] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks." in *INTERSPEECH*, vol. 237, 2011, p. 240.

[9] C. Plahl, R. Schlüter, and H. Ney, "Improved acoustic feature combination for lvcsr by neural networks." in *INTERSPEECH*, 2011, pp. 1237–1240.

[10] C. Plahl, M. Kozielski, R. Schluter, and H. Ney, "Feature combination and stacking of recurrent and non-recurrent neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6714–6718.

[11] K. Kilgour, T. Seytzer, Q. 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*, 2013.

[12] The National Institute of Standards and Technology, "NIST Open Keyword Search 2013 Evaluation (OpenKWS13)," http://www.nist.gov/itl/iad/mig/openkws13.cfm, Apr. 2013, last accessed: July 3, 2013.

[13] F. Metze, Z. A. Sheikh, A. Waibel, J. Gehring, K. Kilgour, Q. B. Nguyen, and V. H. Nguyen, "Models of tone for tonal and non-tonal languages," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 261–266.

[14] K. Laskowski, M. Heldner, and J. Edlund, "The fundamental frequency variation spectrum," *Proceedings of FONETIK 2008*, pp. 29–32, 2008.

[15] J. Gehring, Q. B. Nguyen, F. Metze, and A. Waibel, "Dnn acoustic modeling with modular multi-lingual feature extraction networks," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 344–349.

[16] M. Wölfel, J. W. McDonough, and A. Waibel, "Minimum variance distortionless response on a warped frequency scale." in *INTERSPEECH*, 2003.

[17] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[18] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[19] K. Schubert, "Pitch tracking and his application on speech recognition," Ph.D. dissertation, Diploma Thesis at University of Kalsruhe (TH), Germany, 1998.

[20] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

[21] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[22] S. Stüker, K. Kilgour, and F. Kraft, "Quaero speech-to-text evaluation systems," in *High Performance Computing in Science and Engineering '11*. Springer Berlin Heidelberg, 2012, pp. 607–618.

[23] 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*, 2013.

[24] H. Soltau, F. Metze, C. Fugen, and A. 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.

[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," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010, oral Presentation.

Using Language Adaptive Deep Neural Networks for Improved Multilingual Speech Recognition

Markus Müller, Alex Waibel

Karlsruhe Institute of Technology
Karlsruhe, Germany
m.mueller@kit.edu

Abstract

Building Large Vocabulary Continuous Speech Recognition (LVCSR) systems for under-resourced languages is a challenging task. While plenty of data is available for English, many other languages suffer from a lack of data. There are different methods for tackling this challenge. One possibility is to use data from different languages to boost the performance of a system for a particular target language. With the emerging of LVCSR systems using neural networks (NNs), many research groups have demonstrated the benefits from using additional data in order to improve the system performance. In this work, we propose a method for providing the language information directly to the network, thus enabling it to become language adaptive. We demonstrate the effectiveness of our approach in a series of experiments.

1. Introduction

With the emergance of Deep Neural Networks (DNNs) in the field of automatic speech recognition, different methods have been explored to improve the performance of Large Vocabulary Continuous Speech Recognition (LVCSR) systems. Although DNNs improve the overall system performance, they require a rather large amount of training data to produce reasonable results.

While there are plenty of resources available for English, this does not necessarily hold true when building a system for another language. One possible solution for this problem is to use data from other languages if there is only a limited amount of data available for a particular target language. Several methods have been explored to make use of multilingual data during system training. By using additional data sets, it is for instance possible to either reduce the training time [1] or decrease the word error rate (WER) [2].

Our proposed method aims towards making better use of the provided multilingual data by explicitly providing a language code to the DNN. By doing so, the DNN becomes aware of the different languages used and is able to implicitly learn language specific features. The resulting DNN is language adaptive (LA-DNN) as it processes the language information in addition to the other input features. We evaluate our proposed method by using different ways of adding the language information to the training pipeline.

This paper is structured as follows: In section 2 we review work related to our experiments. In the following section 3 we describe our proposed method for the network training. Section 4 explains our experimental setup and in section 5 we describe and evaluate the results. Finally, we conclude our paper with section 6 where we review our proposed method and also point towards future work.

2. Related work

Current state-of-the-art speech recognition systems rely on using NNs. The networks are being used in various components like audio pre-processing, language modelling and acoustic modelling. In this work, we concentrate on the use of NNs as a part of the audio pre-processing pipeline and the acoustic model.

2.1. Deep Belief Botteneck Features

Deep Belief Neural Networks (DBNFs)[3] process audio features which were extracted from the raw audio using common approaches like mel-scaled cepstral coefficients (MFCC) or logarithmic mel-scaled spectral coefficients (lMel). DBNFs are feed forward neural networks featuring multiple hidden layers. We first pre-train the network using de-noising auto-encoders [4]. This step initializes the network parameters and guides them into an appropriate range. In the next step, the parameters are fine-tuned using Stochastic Gradient Descent (SGD) [5] with mini-batch updates. For the extraction of the features, the layers after the bottleneck are discarded and the output of the bottleneck layer is used as features.

2.2. Multilingual DBNFs

Since neural networks are good at learning different tasks [6], DBNFs can be trained using multiple languages. Furthermore, [7] has shown that the pre-training step is language independent. Therefore it is possible to increase the performance of the network by using the combined data from multiple languages for training the network. After pre-training, the network is fine-tuned. There are two possibilities to deal

with multiple languages at this stage. It is possible to use a merged phoneme set [8] or share the hidden representations among different languages but use language specific output layers ([9], [10], [11], [12]), see figure 1.

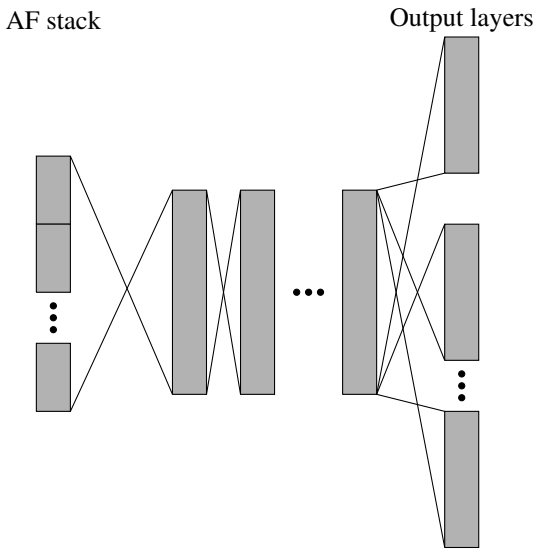


Figure 1: Neural network featuring shared hidden layers and multiple output layers. In our setup, each output layer corresponds to a language specific phone set.

2.3. Augmented Input Features for Neural Networks

Recent publications show that augmenting the input features of the network with additional information like i-Vectors [13] or Bottleneck Speaker Vectors (BSV) [14] increases the overall performance of the system. By providing this additional information, the network adapts to different speakers or acoustic conditions [15]. Since neural networks can process multimodal input data, adding additional information to the features is possible. By doing so, we can provide additional cues to the network. While this was done in the past to provide information about speakers or channels, but, to our knowledge, the use of language codes for building systems in a multilingual environment has not been investigated.

3. Language Adaptive Deep Neural Networks

Augmenting input features with additional information increases the performance of neural networks. Here, we present our approach to add language codes during neural network training in a multilingual environment. By providing this language information in addition to the acoustic features, the network is able take advantage of the language information. As multilingual data can boost the performance of a system when little or no data from the target language is available, we show that this boost can be increased through a language code. As for this code, we chose to encode the language information using 1-of-N coding. This results in a feature vector with one dimension per language.

As pointed out in the related work section, there are multiple possibilities to add data from additional languages throughout the network training. One possibility is to directly merge the available data sets: Create a unified phoneme set, join the different dictionaries and use the audio data jointly to train the system. Another possibility is to build systems for each language individually and then use the individual systems to create language dependent training data for the NNs. It is then possible to share the hidden representations and use language specific output layers. This training technique can be applied to both DBNFs and Hybrid systems. In the latter, the Gaussian Mixture Models (GMMs) are being replaced using a NN.

The language code can be added to the training process of each network. Figure 2 shows the different positions where the language code can be added. It is possible to do an early fusion by appending the language code to the stacked feature frames from the audio pre-processing. Doing so would help the network to discriminate between different languages, but as we will see, this might not be beneficial in all cases. Performing a late fusion is also possible by augmenting the stacked bottleneck features with the language code.

4. Experimental Setup

We conducted a series of experiments in order to assess the performance of our approach. The question is how to augment the existing features with the language code. For training our systems, we use a speech corpus consisting of recordings from Euronews[1], a TV news station [16]. It consists of approximately 70h of acoustic training data per language, sampled at 16 kHz. We use data from 6 languages (English, French, German, Italian, Russian and Turkish), as shown in Table 1. For testing, we used 1.1h of English TV reports.

The pronunciation dictionaries were automatically created using MaryTTS [17]. We selected the languages based on the availability of both recordings from Euronews and pronunciations from MaryTTS. We built the systems using the Janus Recognition Toolkit (JRTk) [18] which features the IBIS decoder [19]. The neural networks were trained using a setup based on Theano ([20], [21]).

Language	Audio Data	# Phonemes
English	72.8h	36
French	68.1h	32
German	73.2h	41
Italian	77.2h	31
Russian	72.2h	27
Turkish	70.4h	31
Total	433.9h	43

Table 1: Overview of used datasets

[1] www.euronews.com

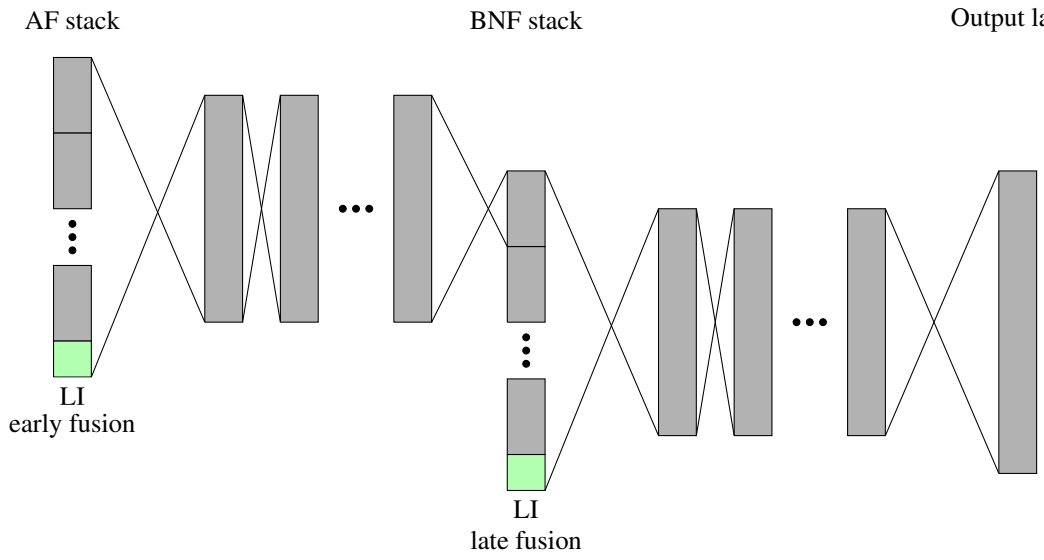


Figure 2: Overview of the network architecture used in our setup. Starting with stacking the acoustic input features (AF), we augment them with a language information (LI) code before feeding them into a DBNF in order to extract BNFs. The BNFs are being stacked as well and the LI code is added. The second DNN computes the phoneme posteriors.

4.1. System Training

For building an initial system, we use a flatstart approach to bootstrap the acoustic models. The audio is pre-processed using 13 dimensional lMEL input features with Δ and $\Delta\Delta$ coefficients which are computed over a window of 16ms that is shifted with 10ms over the audio recording. Based on this inital system, we built a context-dependent system using 6000 context-dependent states. Preliminary experiments have shown that a system using 6000 states has reasonable performance given the amount of available training data.

4.2. DBNF Training

Based on this initial context-dependent system, we extracted samples for training the DBNF network. For training the network, we extracted the samples using a combination of lMel, fundamental frequency variation (FFV) [22] and pitch [23] acoustic features. For the extraction of FFV and pitch, we use a window size of 32ms. The use of additional tonal features has lead to improvements in combination with NNs, even for non-tonal languages such as English [24]. The input features are being stacked using a context of 6 on each side. This results in 13 stacked feature frames being fed into the network at each time step. These stacked frames are then optionally augmented by our 6 dimensional language code which indicates the current language.

The network is layer-wise pre-trained using de-noising auto-encoders. It consists of 5 hidden layers with 1000 neurons per layer. The bottleneck is a narrow layer with only 42 neurons. For fine-tuning, we use stochastic gradient descent with new bob scheduling and log-linear regression. Based on the features extracted by this network, we trained another GMM/HMM system.

4.3. Hybrid System Training

We use the BNF GMM/HMM system to extract a new set of samples for training a DNN. For training this network, we stack features with a context of 7 BNF-frames in each direction, resulting in a total context of 15 frames being fed into the network. This network features 6 hidden layers with a size of 1600 neurons per layer. We use this network as a replacement for the GMMs to estimate the phoneme posterior probabilities. Similar to the training of the DBNF, the input vector for this network is optionally augmented with the language code.

4.4. Merged Phoneme Set

In the first set of experiments, we built a system with language independent models. For training this system, we merged the different training data sets and the pronunciation dictionaries. As we used MaryTTS for generating the dictionaries, we did not need to do a phoneme conversion between the different languages, as all the phonemes already originate from the same phoneme set.

The baseline GMM/HMM system is bootstrapped using all available acoustic data from the 6 languages. This results in 433h of training data for the acoustic model of the system. Based on this initial system, we follow the training procedure described in order to build the BNF based system and the Hybrid system. In order to reduce the training time, we limited the amount of data for the neural network training to 30h per language. To obtain this subset of 30h, we selected a subset of TV reports randomly.

4.5. Language Dependent Phoneme Sets

In a second set of experiments, we built systems with language specific phoneme sets. We used monolingual systems for the extraction of training data for the BNF. Based on this data, we trained a multilingual DBNF by training the hidden layers jointly over all languages and using separate output layers for each language. Based on the multilingual BNF, we again trained systems monolingual for all languages and used them to extract the training data for the Hybrid systems. As for the Hybrid systems, we employed the same training strategy by sharing the hidden layers among languages. The language code was appended to the stacked BNFs.

5. Results

The results section is divided into three different parts. First we present the results from the systems with the merged phoneme set. Next, we show the results from the systems with language specific phoneme sets. This section concludes with a comparison between the multilingual systems and a system trained monolingually.

5.1. Merged Phoneme Sets

The initial GMM/HMM system with a merged phoneme set features a WER of 26.3% as displayed in Table 2. This is rather high, but expected for this type of system: GMM/HMM systems tend to have a poor performance when trained in this multilingual fashion. Using bottleneck features decreases the WER to 21.7% without the language information and 21.2% when adding the language code. The system with the LA-DNN is by 2.4% relative better compared to the system without that additional information. This trend continues for the Hybrid systems. The use of the language information results in a total relative gain of 9.0%. Using a merged phoneme set, adding the language code at both stages (early and late fusion) of network training is beneficial.

System	Baseline	LA-DNN	rel. gain
GMM/HMM	26.3%	26.3%	-
BNF	21.7%	**21.2%**	2.4%
Hybrid	19.3%	**17.7%**	9.0%

Table 2: Overview of results for systems with a merged phoneme set, showing WERs.

5.2. Language Dependent Phoneme Sets

The baseline system with a language dependent phoneme set for English has a WER of 18.9% (see Table 3). This is to a great extend better compared to the system with a merged phoneme set. It is interesting to see that the system with bottleneck features does not benefit from the language code: Providing the language code to the network results in a WER of 18.7%, while the WER is 17.5% when training it without the language information. We therefore use the system without the language code (and the better performance) to write samples for training both Hybrid systems. Based on the performance of the Hybrid systems, it can be seen that adding the language code at the bottleneck layer helps improving the system by 3.5% relative: The system with the language information has a WER of 14.4%, compared to 14.9% WER to the system without. Here, only the late fusion approach leads to improvements.

System	Baseline	LA-DNN	rel. gain
GMM/HMM	18.9%	18.9%	-
BNF	**17.5%**	18.7%	-6.4%
Hybrid	14.9%	**14.4%**	3.5%

Table 3: Overview of results for systems using separate phoneme sets per language, showing WERs.

5.3. Comparison to Monolingual Systems

In a final set of experiments, we compared the performance of monolingual systems to the best multilingual systems. As shown in Table 4, the multilingual systems outperform the systems trained on only one language. Although the relative gain for the hybrid systems (1.4%) decreases compared to the systems using only BNFs (6.3%), we still achieve an improvement by augmenting the input features with the language code.

System	Monol.	ML EN P. Set	rel. gain
GMM/HMM	18.9%	18.9%	-
BNF	18.6%	**17.5%**	6.3%
Hybrid	14.6%	**14.4%**	1.4%

Table 4: Overview of results using language dependent output layers of the neural network, showing WERs.

6. Conclusion

We have presented a method for improving the performance of NN based systems for LVCSR by augmenting the acoustic input features with a language code in a multilingual setup. Gains can be seen throughout different conditions. Depending on the condition, the addition of the language code at either an early and/or a later stage shows the biggest improvements. With the addition of the language information, the DNN becomes language adaptive and is able to better learn the characteristics of different languages.

With our proposed method, the LA-DNN is able to exploit the training data from different languages in a more efficient manner. One of the next steps is to find a replacement for the explicitly coded language information and to auto-

matically extract the language information from the training material in a way similar to i-Vectors or BSVs.

7. References

[1] S. Stüker, M. Müller, Q. B. Nguyen, and A. 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] Q. B. Nguyen, J. Gehring, M. Müller, S. Stüker, and A. Waibel, "Multilingual shifting 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] 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.

[4] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting Deep Bottleneck Features Using Stacked Auto-Encoders," in *Proceedings of the ICASSP*, Vancouver, Canada, May 2013.

[5] L. Bottou, "Stochastic gradient learning in neural networks," *Proceedings of Neuro-Nımes*, vol. 91, no. 8, 1991.

[6] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[7] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proceedings of the Spoken Language Technology Workshop (SLT), 2012 IEEE*, IEEE. IEEE, 2012, pp. 246–251.

[8] N. T. Vu, W. Breiter, F. Metze, and T. 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.

[9] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of Deep-Neural networks," in *Proceedings of the ICASSP*, Vancouver, Canada, 2013.

[10] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the use of a multilingual neural network front-end," in *Proceedings of the Interspeech*, 2008, pp. 2711–2714.

[11] 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.

[12] K. Vesely, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proceedings of the Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 336–341.

[13] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 55–59.

[14] H. Huang and K. C. Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for dnn-based speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4610–4613.

[15] Y. Miao and F. Metze, "Distance-aware dnns for robust speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] R. Gretter, "Euronews: a multilingual benchmark for asr and lid," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[17] 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.

[18] M. Woszczyna, N. Aoki-Waibel, F. D. Buø, N. Coccaro, K. Horiguchi, T. Kemp, A. Lavie, A. McNair, T. Polzin, I. Rogina, C. Rose, T. Schultz, B. Suhm, M. Tomita, and A. Waibel, "Janus 93: Towards spontaneous speech translation," in *International Conference on Acoustics, Speech, and Signal Processing 1994*, Adelaide, Australia, 1994.

[19] H. Soltau, F. Metze, C. Fugen, and A. 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.

[20] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[21] 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, oral Presentation.

[22] K. Laskowski, M. Heldner, and J. Edlund, "The Fundamental Frequency Variation Spectrum," in *Proceedings of the 21st Swedish Phonetics Conference (Fonetik 2008)*, Gothenburg, Sweden, June 2008, pp. 29–32.

[23] K. Schubert, "Grundfrequenzverfolgung und deren Anwendung in der Spracherkennung," Master's thesis, Universität Karlsruhe (TH), Germany, 1999, in German.

[24] F. Metze, Z. Sheikh, A. Waibel, J. Gehring, K. Kilgour, Q. B. Nguyen, V. H. Nguyen, *et al.*, "Models of tone for tonal and non-tonal languages," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 261–266.

Punctuation Insertion for Real-time Spoken Language Translation

Eunah Cho, Jan Niehues, Kevin Kilgour, Alex Waibel

Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology, Germany
{eunah.cho|kevin.kilgour|jan.niehues|alex.waibel}@kit.edu

Abstract

Sentence segmentation and punctuation insertion in the output of automatic recognition systems is essential for its readability as well as for the performance of subsequent applications, such as machine translation systems. While a longer context can boost the accuracy of inserted punctuation marks, it drastically increases the delay in the spoken language translation system.

In this work, we investigate the impact of shorter context in punctuation insertion task on simultaneous speech translation system. We suggest a new scheme within stream decoding where the time delay consumed on punctuation prediction is avoided. Our evaluations on the English TED talks show that our suggested scheme can be used as an efficient method to punctuate recognized streams in real-time scenarios. While outperforming a conventional language model and prosody based punctuation prediction system, our model maintains a comparable performance compared to systems that require longer contexts.

1. Introduction

Inserting reliable punctuation marks and sentence segmentation into automatically recognized transcripts plays an important role in spoken language translation (SLT) systems. Many of the conventional automatic speech recognition (ASR) systems generate either no or unreliable punctuation marks. Without a proper punctuation insertion component, therefore, the automatically recognized output is hard to read for humans. Also, it affects the performance when the ASR output is used in subsequent applications of natural language processing (NLP), such as machine translation (MT) systems. Missing proper punctuation marks especially degrades the performance of MT systems, since most of them are trained using well-structured texts, such as news corpus, where sentence boundaries are clear and well-formed.

One of the commonly used methods for inserting punctuation marks into the ASR output is the language model (LM) and prosody based scheme as discussed in [1]. It has an advantage that it incorporates acoustic features keeping the process relatively fast. Recently, punctuation insertion models using a monolingual translation system [2, 3] have shown the effectiveness in improving the performance of MT systems when they are applied to the ASR output. A monolingual translation system is an MT system which translates non-punctuated input text into punctuated text. The conventional monolingual translation system suggested in previous work uses overlapping window for input. Since it can provide a very long context, a great performance improvement on the MT for ASR outputs can be achieved using this technique. Overlapping windows, however, make the system difficult to be used in real-time scenarios without long latencies.

One indisputably crucial aspect in inserting punctuation marks for real-time speech translation is the time delay. Longer context is preferred for better prediction performance but it causes more delay.

In this paper, we suggest an efficient punctuation insertion scheme for real-time SLT systems, using the monolingual translation system. Our punctuation insertion and sentence segmentation system is designed to take the output of a stream decoding ASR system. The input to the monolingual translation system is modified so that latency can be decreased while maintaining similar translation performance. We performed experiments both on audio streams as well as manual transcripts, in order to give in-depth analysis on the impact of different length of context in the punctuation insertion scheme.

This paper is organized as follows. In Section 2, a brief overview of past research on punctuation insertion for varied scenarios is given. The task of inserting punctuation marks for real-time translation systems and its related challenges are discussed in Section 3. Section 4 describes how we model the punctuation insertion system for real-time speech translation scenario. The systems we used throughout this work are described in detail in 5. Section 6 shows our experimental setups and results, followed by Section 7 where we conclude our discussion.

2. Related Work

In previous work [4], sentence segmentation for ASR output was modeled based on LM probabilities and prosody. The authors emphasized that choosing a proper segment length for the different MT systems boosts the translation performance. In this work, commas and final periods are not considered separately, but together in order to form segment boundaries. A threshold was used to control the average number of segments per sentence.

Recently MT-driven approaches have emerged as an effective method to insert punctuation marks in ASR output. An approach using a modified phrase table was introduced in [5] as a method to restore commas. Sentence boundaries are generated based on a decision tree on the source side. Applied to three different language pairs, their method significantly improved translation performance.

Among different MT-driven techniques to model punctuation marks for spoken language, the monolingual translation system [3, 2] has shown an outstanding performance in improving machine translation quality in evaluation campaigns [6, 7]. Using this system, a non-punctuated source language is translated monolingually into punctuated source language. In [3], authors made in-depth analysis on three different approaches to restore punctuation marks using an MT system. Among the three systems, they achieved their best performance when using the translation system to translate non-punctuated text into punctuated one. In their work, however, it was assumed that reliable sentence boundaries are already given. Therefore, punctuation marks within each of the given sentence boundaries are restored.

Based on the work in [3], authors in [2] extended the system so that sentence boundaries can also be predicted. In order to model the possibility to insert a final period everywhere given a segment, they randomly cut the training data for the monolingual translation system. Also, the test data was prepared with a shifting window of 10 words.

While the work mentioned above focused on enhancing punctuation accuracy or the machine translation performance when using the punctuated ASR output, the authors in [8] made an extensive study on different segmentation strategies and latency. They inserted segments based on various techniques into ASR output for real-time translation experiments. It was shown that a good performance can be achieved when they use the conjunction-based segmentation strategy along with a comma-based segmentation.

The input segment length and machine translation quality are studied in [9]. In this work, a statistical machine translation (SMT) decoder which processes a continuous input stream was suggested. Using the decoder they achieved improved translation quality at relatively low latencies.

3. Real-time Spoken Language Translation

In order to be useful a real-time spoken language translation system has to, among many other challenges, deal with the problem of latency. The latency of a real-time spoken language translation system is the time between when a word is spoken and when its transcription and translation are displayed to the user [1]. If the latency is more than a few seconds then the whole translation system becomes unusable and frustrating for the user. Each component adds to the latency, due to computation time, communication time and required future context.

Communication time can be kept to a minimum by having a fast connection and low overhead between the individual components. Computation time may be reduced by running the components on fast servers with multiple cores and by parallelizing those parts of the individual components that can be. It may also require sacrificing accuracy by using smaller faster models.

In order to reduce the apparent latency the speech recognition component can be configured to output its current best hypothesis about once a second. The displayed output is then often updated by a newer, possibly better, hypothesis. This type of setup has a much higher user acceptance than the alternative setup where the speech recognition component waits until it has a stable hypothesis before outputting it which can sometimes result in 8 or more words appearing at once.

The MT component is even more dependent on context than the speech recognition component and often has to wait for the whole sentence to be recognized before it can be properly translated. A fast enough MT system can re-translate the sentence each time the ASR system recognizes a new word and change the output displayed to the user. For this to work, however, the MT system requires the ASR output to be segmented into proper sentences.

These design decisions for both the ASR component, the MT component and the real-time spoken language translation system as a whole pose some significant challenges for the punctuation prediction component that converts the stream text output stream of the ASR component into proper sentences required for the MT system. A major side affect of the ASR component constantly updating its current hypothesis is that the punctuation prediction component has to deal with possibly changing inputs. It also has to have a fast computation time because the ASR system is sending updates very frequently. As the MT component requires sentence boundary information as soon as possible in order to function properly the punctuation prediction component has function well with only very little future context.

Although the monolingual translation system [2] shows a good performance in the subsequent application, adopting this system for the real-time speech translation system causes an unacceptable amount of latency due to its long shifting window of 10 words. This component alone would introduce more latency into the whole system than the desired total average latency.

4. Model

In order to decrease the delay in the real-time speech translation system, we use a streaming input scheme instead of the overlapping window. In this section, we describe how the streaming input scheme works.

Our in-house stream decoding ASR system stores its recognition in two separate stacks. In one stack it saves its final 1-best list for words $w = \{w_l, \ldots, w_m\}$. Their following words are stored in another stack $v = \{v_{m+1}, \ldots, v_n\}$, which is not the final recognition yet. Since this stack v is flexible depending on the upcoming context, it is updated

based on the context and whenever it is updated, the changes are shown to users.

In our punctuation insertion setup, we introduce another stack for recognized words before w, in order to consider more context. The history stack h is defined as:

$$h = \{h_{l-c}, \ldots, h_{l-1}\} \quad (1)$$

The context c is chosen as four throughout this work. When there are fewer previous words available in the initial part of the recognition, only upto available context is used.

The newly punctuated string is then obtained by

$$w' = m(h + w) \quad (2)$$

where m denotes the monolingual translation system. Its scheme will be described in detail in Section 5.3. Parts of the generated output is taken as the final string.

$$s = \{w'_{l-c}, \ldots, w'_{m-4}\} \quad (3)$$

At the same time the history stack is updated.

$$h = \{w'_{m-3}, \ldots, w'_{m}\} \quad (4)$$

Thereby we input punctuated text into the monolingual translation system and repunctuate it. Although this leads to a slight mismatch between training and testing data, using this way we can guarantee punctuation can be inserted when the longest context is available.

Table 1 shows how an excerpt from an automatically recognized transcript is punctuated in our monolingual translation system scheme. History stack is marked in blue box.

Input	OK but then after a while
Output	OK. But then, after a while,
Input	then, after a while, I realized this is
Output	then, after a while, I realized this is
Input	I realized this is my life this is six months of
Output	I realized this is my life. This is six months of
Input	is six months of my life and
Output	is six months of my life. And
Input	of my life. And this . . .
Output	of my life. And this . . .

Table 1: History stack and punctuation output

For the non-final ASR recognition stack v, we generate the possible output string $m(h + v)$ and show it to users.

An advantage of this model is that while longer history is utilized, the decision on punctuation insertion on the current window can be made instantly, minimizing the time delay consumed on sentence segmentation. Also, by supporting the stream decoding, users can see the updating recognition as well as its most probable punctuation marks within.

and I said, “OK, it ’s the huge file. OK,
I said, “OK, it ’s the huge file. OK, but
said, “OK, it ’s the huge file. OK, but then
OK. it ’s the huge file. OK, but then, after
it ’s the huge file. OK, but then, after a
’s the huge file. OK, but then, after a while,
the huge file. OK, but then, after a while, I
huge file. OK, but then, after a while, I realised
file. OK, but then, after a while, I realised this
OK , but then, after a while, I realized. this is
but then, after a while, I realized. this is my
then , after a while, I realized. this is my life.
after a while, I realized. this is my life. this
a while, I realized. this is my life. this is
while I realized. this is my life. this is six
I realized. this is my life. this is six months
realized . this is my life. this is six months of
this is my life. this is six months of my
is my life. this is six months of my life
my life. this is six months of my life, and
life . this is six months of my life, and this
this is six months of my life. and this fire
is six months of my life. and this fire. so
six months of my life. and this fire. so, I
months of my life. and this fire. so I was
of my life. and this fire. so I was a
my life. and this fire. so I was a little
life , and this fire. so I was a little bit
and this fire. so I was a little bit skeptical
this fire. so I was a little bit skeptical of

Table 2: Output of monolingual translation system with overlapping window of 10

As a comparison, Table 2 shows the actual output of monolingual translation system with overlapping input, for the same segments shown in Table 1. Since the system is using an overlapping window of 10 words, each encountering word (marked in red box) has to be translated 10 times as well. In this overlapping window system [2], each token is translated ten times and a punctuation mark is inserted depending on how often it occurs after this token. For example, augmenting a punctuation mark after the first encountering word *OK* needs previous ten translations.

From the comparison between Table 1 and Table 2, we can observe that the streaming segmentation system can decrease the latency introduced by using the monolingual translation system with overlapping window. While the suggested streaming segmentation will punctuate the given segments in only 5 times of translation, using the overlapping window requires 30 times of translation.

5. System Description

In this section, we discuss the systems we use throughout this work. The English audio data is decoded using our ASR system. A brief description on our LM and prosody based segmenter, which is used as one of the baseline systems, is also given. Once the punctuation marks are inserted using different segmentation strategies, we translate this test data into German in order to measure the performance of the real-time punctuation insertion system.

5.1. English ASR System

The speech recognition is performed using in-house decoder in an online setup. Using a framesize of 32ms and a frameshift of 10ms the audio stream is converted in a stream of 40 dimensional lMel feature vectors.

The hybrid DNN/HMM acoustic model uses a context dependent quinphone setup with three states per phoneme, a left-to-right HMM topology without skip states. The neural network has in input window of ± 6 frames leading to an input layer size of 520 neurons, this is followed by 4 layers of 2k neurons and a finial classification output layer containing just over 8k neurons.

The neural network is pretrained layerwise using denoising autoencoders with a 20 million mini batches. After pretraining the final layer is added, with the output layer using the softmax activation function. The full DNN is then fine-tuned using the newbob learning rate schedule. All training is performed using Theano [10] on the TED [11] and Quaero data [12].

For the language model training texts from various sources such as webdumps, scraped newspapers and transcripts are used. The 120k vocabulary is selected by building a Witten-Bell smoothed unigram language model using the union of all the text sources vocabulary as the language models' vocabulary (global vocabulary). With the help of the maximum likelihood count estimation method described in [13] we found the best mixture weights for representing the tuning set's vocabulary as a weighted mixture of the sources word counts thereby giving us a ranking of all the words in the global by their relevance to the tuning set.

Using this vocabulary language models are built from each of the sources and interpolated using the SRILM toolkit [14] so as to maximally reduce the perplexity of the tuning set.

5.2. LM and Prosody based Segmentation

The language model and prosody based segmenter employs a 4-gram language model trained on punctuated text. In order to predict punctuation marks a context of four words, two prior and two after the possible punctuation mark, is taken into consideration.

The language model is used to calculate three scores. The first one is the score without an inserted punctuation mark as

$$P(w_{i-1}, w_i, w_{i+1}, w_{i+2}) \tag{5}$$

while the second one is the score with a comma.

$$P(w_{i-1}, w_i, \texttt{@COMMA}, w_{i+1}, w_{i+2}) \tag{6}$$

The last one is calculated by followings.

$$P(w_{i-1}, w_i, \texttt{@STOP}, w_{i+1}, w_{i+2}) \tag{7}$$

A dynamic scaling factor is applied to the punctuation mark scores in order to prevent both very short sentences and very long sentences. In parallel to the language model a prosody component searches for pauses over t_θ seconds and then force terminates any sentences.

5.3. Monolingual Translation System

Monolingual translation system for punctuating English data is trained on the English side of the European Parliament data, News Commentary, TED[1], and the common crawl corpus.

As a preprocessing step, the noisy part of the common crawl data is filtered out using an SVM model as described in [15]. After preprocessing is applied, the normalized training data is resegmented randomly so that punctuation marks can be observed in all possible locations in each line.

For the source side of the training, we removed final period, comma, question mark, and exclamation mark. Double quotation marks are also removed as they are relatively frequent in TED talks. In addition to processing the punctuation marks, we also lowercased every single word on the source side. Since automatically recognized words often miss correct case information, we aim to restore the case information altogether with punctuation marks using this one system. Altogether the training data consists of 10.1 million English words.

The Moses package [16] is used to build the phrase table. We build a 4-gram language model on the entire punctuated target side using the SRILM Toolkit [17]. Word alignment is learned automatically using the GIZA++ Toolkit [18]. A bilingual language model [19] is used along with a 9-gram part-of-speech (POS)-based language model. The POS is learned from TreeTagger [20]. In addition to this POS-based language model, we train a $1,000$ class cluster [21] and use the cluster codes for the additional 9-gram language model. The model weights were optimized on the official test set of IWSLT evaluation campaign in 2012.

5.4. English-German MT System

For evaluating our online punctuation insertion schemes, we translate the testsets with different segmentation and punctuation marks into German. For the translation, we use online

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

English to German phrase-based translation system. The system is trained on the parallel corpus of Europarl, News commentary, TED, and the noise-filtered common crawl data. For the monolingual data we take the News Shuffle corpus. Detailed statistics on corpus can be found in [7].

We build a 4-gram language model on the German side of TED data which is used as an in-domain language model. In addition to this language model, we used a bilingual language model on all available parallel data as described in [19]. Also, we used a 4-gram language model that is built based on cross entropy with the development data. For the in-domain TED data, we applied the cluster algorithm [21]. Once the TED data is clustered into $1,000$ classes, we build a 9-gram language model and used it as an additional model.

In order to address the word order difference between English and German, we use the POS-based reordering [22] along with the tree-based [23] and lexicalized reordering rules. For optimization of the log-linear combination of models, we use minimum error rate training [24].

For evaluating differently segmented testsets, we use the Levenshtein minimum edit distance algorithm [25] in order to align hypothesis against the reference translation.

5.4.1. Phrase Table Preparation

For online translation systems, it is impossible to generate a perfectly fitting phrase table for each input data. Therefore, we build a phrase table based on the vocabulary in the training data. In order to decrease the size of the model for online scenario, we first filtered out words which occurred in the corpus less than four times. Also, phrases that are longer than 4-grams are filtered out as well.

6. Experiments and Results

In order to measure the impact of different segmentation methods and models on MT, we experiment on the official test set of IWSLT evaluation campaign 2013. The English manual transcript of this test data has 993 sentences, or 17.8K tokens. The audio is 2h 16m long.

The proposed streaming punctuating prediction (StreamingInput) system is compared to both a low latency baseline language model and prosody based punctuation prediction (LM, Prosody) system as well the high latency but highly accurate monolingual translation (Baseline) system using a 10 word moving window. Table 3 presents these systems' translation performance of the test data. The numbers are reported in case-sensitive BLEU [26].

In the first row, we first show the translation performance when using the simple LM and prosody based segmentation, available only for the ASR output. In the baseline system, both ASR output and manual transcript are punctuated using the conventional monolingual translation system, using overlapping windows, as shown in [2]. Therefore, the shift window is applied so that each word is translated for ten times. As it is not for online scenario, the phrase table is also generated upon the knowledge of the each test data.

Punctuation	ASR Output	Manual Transcript
LM, Prosody	9.74	-
Baseline	11.18	19.57
StreamingInput	**11.55**	**19.41**

Table 3: Translation performance of the proposed system compared to a fast LM, prosody based model as well as a high latency, but highly performant monolingual system using an overlapping window

We can see that when we use the suggested punctuation insertion scheme, we achieve 11.55 BLEU points in the ASR test data, beating the conventional LM and prosody based model by 1.8 BLEU points. Even though this system is using relatively shorter context and the less-fitting phrase table than the traditional monolingual translation system, the translation performance is comparable with the baseline monolingual translation system's. Although the translation was slightly worse when using this system for punctuating the manual transcript, we achieve an improvement of 0.4 BLEU in the ASR translation task which is its intended use case.

Due to the small model footprint and the use of an efficient MT decoder the stream-based punctuation prediction setup incurs only minimal computational cost, comparable to the punctuation model based on LM and prosody without having much future context requirements. This fast system also allows for updated punctuation when new data is received. As this component does not add further communication overhead, the total latency of the real-time speech translation system is not negatively impacted.

7. Conclusions

In this paper, we present a new punctuation insertion scheme for real-time spoken language translation system. Taking streamed input from an ASR decoder, the suggested scheme can improve the output of the speech translation without negatively impacting the speech translation system's latency. The experiments show that our low-latency real-time punctuation insertion system can achieve a comparable performance to an offline system requiring a large context window.

As future work, we intend to evaluate the system performance on further language pairs. We would also like to investigate the possible integration of neural network and conditional random field-based punctuation prediction models.

8. Acknowledgements

The project leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 645452.

9. References

[1] E. Cho, C. Fügen, T. Herrmann, K. Kilgour, M. Mediani, C. Mohr, J. Niehues, K. Rottmann, C. Saam, S. Stker, and A. Waibel, "A real-world system for simultaneous translation of german lectures," in *INTERSPEECH*, Lyon, France, 2013.

[2] 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.

[3] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling Punctuation Prediction as Machine Translation," in *IWSLT*, San Francisco, CA, USA, 2011.

[4] S. Rao, I. Lane, and T. Schultz, "Optimizing Sentence Segmentation for Spoken Language Translation," in *Proc. of Interspeech*, Antwerp, Belgium, 2007.

[5] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence Segmentation and Punctuation Recovery for Spoken Language Translation," in *ICASSP*, Las Vegas, Nevada, USA, April 2008.

[6] 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 International Workshop for Spoken Language Translation (IWSLT 2013)*, Heidelberg, Germany, 2013.

[7] 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 International Workshop for Spoken Language Translation (IWSLT 2014)*, Lake Tahoe, CA, USA, 2014.

[8] V. K. R. Sridhar, J. Chen, S. Bangalore, A. Ljolje, and R. Chengalvarayan, "Segmentation strategies for streaming speech translation," in *HLT-NAACL*, 2013, pp. 230–238.

[9] M. Kolss, S. Vogel, and A. Waibel, "Stream decoding for simultaneous spoken language translation."

[10] 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, oral Presentation.

[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*, 2013.

[12] S. Stüker, K. Kilgour, and F. Kraft, "Quaero 2010 speech-to-text evaluation systems," in *High Performance Computing in Science and Engineering'11*. Springer, 2012, pp. 607–618.

[13] A. Venkataraman and W. Wang, "Techniques for effective vocabulary selection," *arXiv preprint cs/0306022*, 2003.

[14] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh International Conference on Spoken Language Processing*, 2002.

[15] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The kit english-french translation systems for iwslt 2011," in *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT)*, 2011.

[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, Demonstration Session*, Prague, Czech Republic, 2007.

[17] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *ICSLP*, Denver, CO, USA, 2002.

[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] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *WMT*, Edinburgh, UK, 2011.

[20] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[21] 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. [Online]. Available: http://dx.doi.org/10.3115/977035.977046

[22] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *TMI*, Skövde, Sweden, 2007.

[23] 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, Georgia, USA, 2013.

[24] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *WPT*, Ann Arbor, MI, USA, 2005.

[25] 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.

[26] 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.

Class-Based N-gram Language Difference Models for Data Selection

Amittai Axelrod
Johns Hopkins University
and University of Maryland
amittai@clsp.jhu.edu

Yogarshi Vyas, Marianna Martindale, Marine Carpuat
University of Maryland
marine@cs.umd.edu

Abstract

We present a simple method for representing text that explicitly encodes differences between two corpora in a domain adaptation or data selection scenario. We do this by replacing every word in the corpora with its part-of-speech tag plus a suffix that indicates the relative bias of the word, or how much likelier it is to be in the task corpus versus the pool. By changing the representation of the text, we can use basic n-gram models to create *language difference models* that characterize the difference between the corpora. This process enables us to use common models with robust statistics that are tailored to computing the similarity score via cross-entropy difference.

These improvements come despite using zero of the original words in the texts during our selection process. We replace the entire vocabulary during the selection process from 3.6M to under 200 automatically-derived tags, greatly reducing the model size for selection.

When used to select data for machine translation systems, our language difference models lead to MT system improvements of up to +1.8 BLEU when used in isolation, and up to +1.3 BLEU when used in a multi-model translation system. Language models trained on data selected with our method have 35% fewer OOV's on the task data than the most common approach. These LMs also have a lower perplexity on in-domain data than the baselines.

1. Introduction

Data selection is a popular approach to domain adaptation that requires quantifying the relevance to the domain of the sentences in a pooled corpus of additional data. The pool is sorted by relevance score, the highest ranked portion is kept, and the rest of the data discarded. By identifying the subset of the data pool that is most like the in-domain corpus and using it instead of the entire data pool, the resulting translation systems are more compact and cheaper to train and run than the full system trained on all of the available data. The underlying assumption in data selection is that the large corpus likely includes some sentences that fall within the target domain. These in-domain sentences should be used for training. Any large data pool will also contain sentences that are irrelevant at best to the domain of interest. At worse, these sentences that are *so* unlike the in-domain data that their presence makes the downstream models worse, and thus they should be removed from the training set.

We note that the models used for data selection are n-gram language models. These are typically used to characterize an entire corpus. However, the data selection scenario is not a characterization task, but a differentiating one. For every sentence in some large, general data pool of potentially dubious provenance, we would like to compute its relevance to some particular in-domain corpus, regardless of what it contains. One could even claim that we do not care what the in-domain data looks like, we just want more of whatever it is.

This supports the use of different models for selecting the data than for using the data in some downstream application. In particular, during the selection process it is more important to know how the corpora differ than how they are alike. We present a simple method for constructing a discriminative representation of the general corpus, and use it to train a language model that is focused on quantifying the difference between the in-domain and general corpora.

2. Background

2.1. Data Selection

The standard approach for data selection uses *cross-entropy difference* as the similarity metric [1]. This procedure leverages the mismatch between the data pool

and the task domain. It first trains an in-domain language model (LM) on the task data, and another LM on the full pool of general data. It assigns to each full-pool sentence s a *cross-entropy difference score*,

$$H_{LM_{IN}}(s) - H_{LM_{POOL}}(s), \quad (1)$$

where $H_m(s)$ is the per-word cross entropy of s according to language model m. Lower scores for cross-entropy difference indicate more relevant sentences, i.e. those that are *most like* the task *and most unlike* the full pool average. In bilingual settings such as machine translation, the *bilingual Moore-Lewis* criterion [2] combines the cross-entropy difference scores from each side of the corpus; i.e. for sentence pair $\langle s_1, s_2 \rangle$:

$$\begin{aligned}&(H_{LM_{IN_1}}(s_1) - H_{LM_{POOL_1}}(s_1)) \\ &+(H_{LM_{IN_2}}(s_2) - H_{LM_{POOL_2}}(s_2)) \quad (2)\end{aligned}$$

After sorting on the relevant criterion, the top-n sentences (or sentence pairs) are selected to create a task-relevant training set. Typically a range of values for n is considered, selecting the n that performs best on held-out in-domain data.

Cross-entropy difference data selection methods are a common pre-processing step for machine translation applications where model size or domain specificity are important. These methods have been extended within the MT community, *e.g.* by [3] using IBM model socres, edit distance [4], neural language models [5]. Furthermore, [6] showed improvements by using EM to identify true out-of-domain data from the pool to contrast against the in-domain data. They also highlight the distrinction between *relevance* and *fluency* that underlies the proposed language difference models. More recently, [7] proposed abstracting away rare words while training the models used for the selection step.

We present a simple method for modeling the difference between two corpora, one that is tailored to fit existing cross-entropy methods for data selection and can readily be applied to other problems.

2.2. Some Words Matter More

All words in a text do not contribute equally to characterize the text. However, which words are more important than others depends on the application. The most frequent words get higher probability in a normal n-gram language model. In topic modeling, content words are prized for what they convey and stopwords are ignored. By contrast, content words are largely ignored in stylometry when deciding the relevance of a text collection to a particular author or genre. Instead, the relevance is determined using function word and part of speech features together. In particular, [8] uses the difference in word frequencies across authors, genres, or eras. Syntactic structure or at least certain syntactic constructions are a potentially more informative source of stylometric features, [9] and [10]. POS tag sequences were introduced as stylometric features by [11] for document classification. [12] subsequently noted that the frequency of the word should be taken into account, else the classifier learns too much about rare events whose empirical estimates of counts and contexts might be incomplete.

A common thread is abstracting words into classes or groups that have more robust statistics. Sequences of these classes, such as part-of-speech (POS) tags, are then used as lightweight representations of the syntactic structure of a sentence. These can be thought of as a quantifiable proxy for sentence register, style, genre, and other ways of characterizing a corpus. For the more specific task of domain adaptation or data selection, replacing some words in the text with their POS tags is a way of creating general templates of what the text is like. This has been used in MT to build better domain-adapted language models [13] and for broader-coverage data selection [7] as mentioned previously.

3. Proposed Method

The method of [1] for data selection explicitly takes advantage of the inherent difference between the task and the pool corpora. Looking for sentences that are like the task corpus and are unlike the pool does not work if the two corpora are very similar. The language models trained on similar corpora will have similar distributions, so the scores in Equation 1 will subtract to zero.

However, in a domain adaptation scenario, the existence of a substantial difference between the task and pool corpora is axiomatic. If this were not the case, then there would be no adaptation scenario! The cross-entropy difference method exploits this difference between the corpora. Because the corpora must differ, then so must be the language models trained on them. Because the language models must differ, then subtracting the scores finds more relevant sentences.

We perform a similar trick with the text itself: where there is a difference between the language mod-

els trained on the task and the pool, then there is a difference between the frequencies of certain words in the corpora. Where the frequencies of words differ, the corpora differ. Where they do not differ, neither do the corpora, so we can expect to see them at the same rate. We can exploit this difference, because we know we are going to subtract the cross-entropy scores.

Words that appear with approximately the same frequency in both texts will have roughly similar cross-entropies according to both the task and pool language model. These words contribute negligibly to the cross-entropy difference scoring because the Moore-Lewis criterion subtracts the two language model scores. This means that the similarity score is only based on words whose empirical distributions are substantially different from one corpus to the other. These words appear in n-grams whose probabilities also differ between the corpora, and these are the non-zero components of the cross-entropy difference score for the sentence.

Whether the word is common or rare or inherently topical has little bearing on the score: if it appears similarly often in both corpora – regardless of how often that is – it will not contribute to the cross-entropy difference. A word's impact on data selection depends on the two corpora being compared in a specific data selection or domain adaptation scenario.

We can take advantage of this to construct models of the corpora that specifically capture which words matter for computing cross-entropy difference between these specific two in-domain and pool data sets. Rather than build new infrastructure, we will simply construct a representation of the text that captures this discriminative information, and then train an n-gram language model on the new representation. This approach has the advantage of being readily reproducible. We call the resulting model a *l*anguage difference model, and use it to compute the cross-entropy difference scores.

The representation of the text is straightforward: we replace each and every word with a token consisting of two parts: the POS tag of the word, and a suffix indicating how much more likely the word is to appear in the task corpus than in the pool corpus.

We use the *ratio of the word's probabilities* in the corpora to determine how much the two specific corpora differ with respect to a word. The ratio simply divides the frequency of the word in the task corpus by the frequency of the word in the pool corpus. This can also be readily computed using unigram LMs trained on each of the corpora.

In this particular work we distinguish this ratio as being quantized by powers of ten, as shown in Table 1. We also add an eighth suffix ("/low") to indicate words that occur fewer than 10 times, following the results in [7]. This was done to enable direct comparison of the contribution of the skew suffixes with prior work. In general, we only bucketed the probability ratios by powers of ten to demonstrate the potential of language difference models for data selection. There is ample room for exploration.

Frequency Ratio ($\frac{Task}{Pool}$)	Suffix	Example Token
$1000 \leq x$	/+++	JJ/+++
$100 \leq x < 1000$	/++	NNS/++
$10 \leq x < 100$	/+	NN/+
$10^{-1} \leq x < 10$	/0	DET/0
$10^{-2} \leq x < 10^{-1}$	/-	NN/-
$10^{-3} \leq x < 10^{-2}$	/- -	JJ/- -
$x < 10^{-3}$	/- - -	NNP/- - -

Table 1: Suffixes to indicate how indicative a word is of one corpus or the other

Our class-based n-gram language difference model representation condenses the entire vocabulary from hundreds of thousands of words down to 150-190 total types, as shown in Table 2. Each type conveys a class of words' syntactic information –which can be considered a proxy for style – as well as information about how indicative the words are of one corpus or the other.

Language	Vocab (full)	Labels (Task)	Labels (Pool)
English	3,904,187	148	182
French	3,681,086	147	190

Table 2: Corpus vocabulary size before and after replacing all words with discriminative labels

As an example, consider the word *supermassive*, which appears 21 times in the in-domain corpus, and 35 times in the data pool. The task pool contains 4.2M tokens, and the data pool contains 1,180M tokens. The empirical frequency ratio:

$$\frac{C_{task}(supermassive)}{4.2M} \div \frac{C_{pool}(supermassive)}{1,180M}$$

is calculated by:

$$\frac{1,180M}{4.2M} \times \frac{C_{task}(supermassive)}{C_{pool}(supermassive)} \approx 281 * \frac{21}{35} = 169$$

The derivation of the labels used to replace a phrase such as *supermassive black holes* in the class-based language difference representation used for data selection is shown in Table 3.

words:	supermassive	black	holes
POS:	JJ	JJ	NNS
ratio:	$100 \leq 169 < 1000$	$10^{-1} \leq 8 < 10$	$10 \leq 28 < 100$
label:	JJ/++	JJ/0	NNS/+

Table 3: Deriving the discriminative representation of a phrase. Only the tokens in the last line appear in the language difference model, as they are 1-to-1 replacements for the original words in first line.

Once the text has been transformed into the class-based language difference representation, we proceed with the standard cross-entropy difference algorithm. After computing the similarity scores and using them to re-rank the sentences in the pool corpus, we transform the text back into the original words and train the downstream LMs and SMT systems as normal. This process enables us to use models with robust statistics for how the corpora differ in order to compute the relevance score, and then use the traditional, n-gram based systems for the downstream MT pipeline.

4. Experimental Framework

Our experiments were based on the French-to-English MT evaluation track for IWSLT 2015. The task domain was defined to be TED talks, a translation subdomain with only 207k parallel training sentences. The data pool consisted of 41.3M parallel sentences from assorted sources, described in Table 4. The parallel Wikipedia and TED corpus were from the ISWLT 2015 website.[1] The remaining corpora were obtained from WMT 2015.[2] Our systems were tuned on *test2010* and evaluated using BLEU [14] on *test2012*, and *test2013* from the same TED source.

All parallel data was tokenized with the Europarl tokenizer[3] and lowercased with the `cdec` tool. We found it was necessary to further preprocess the data by using perl's Encode module to encode as UTF-8 octets and decode back to characters. We replaced fatally malformed characters with the Unicode replacement character, `U+FFFD`.

Dataset	# of sentences
Europarl v7	2.0M
News Commentary	0.2M
Common Crawl	3.2M
10^9 Fr-En	22.5M
UN Corpus	12.8M
Wikipedia	0.4M
TED corpus	0.2M

Table 4: Provenance of the 41M sentence French-English data pool.

We trained all SMT systems using cdec [15], tuned with MIRA [16]. The (4-gram) language models used for the selection process were all trained with KenLM [17]. The Stanford part-of-speech tagger [18] generated the POS tags for both English and French.[4]

5. Results and Discussion

The standard Moore-Lewis data selection method uses normal n-gram language models to compute the cross-entropy scores according to each of the task and pool language models. These scores get subtracted into the cross-entropy difference score that is used to rank each sentence in the data pool. We have proposed computing these cross-entropy values differently: using a language model trained over the class-based language difference labels for each word in the sentence, instead of the LM trained on the words themselves.

As an experimental baseline, we perform Moore-Lewis data selection in the standard way using the normal text corpora (`"xediff"`). This method is shown in grey in all figures. The language models were standard n-gram word-based models, with order 4 and the vocabulary fixed to be the pool lexicon minus singletons, plus the task lexicon. The final English-side vocabulary contained 1,796,862 words, and the French side 1,728,231, with the size reflecting the noisiness and

[1] `https://sites.google.com/site/iwsltevaluation2015/data-provided`

[2] `http://www.statmt.org/wmt15/translation-task.html`

[3] `http://www.statmt.org/europarl/v7/tools.tgz`

[4] The Stanford NLP tools use the Penn tagsets, which comprise 43 tags for English and 31 for French.

heterogeneity of the data pool.

For a slightly harder baseline, we compare against the approach of [7] in WMT 2015, who replace all rare words (count < 10) with their POS tag during the selection process (`"min10"`). This method is shown in dark blue in all figures. This baseline is expected to provide a modest improvement in translation quality and a large improvement in lexical coverage. Finally we perform our proposed method of language difference models, replacing all words in the corpora with a class-based difference representation during the selection process (`"new"`). These results are shown in orange.

Each of these variants produces a version of the full pool in which the sentences are ranked by relevance score. For each of those ranked pools, we evaluate language models trained on increasingly larger *slices* of the data ranging from the highest scoring $n = 500$K to the highest scoring $n = 5$M sentence pairs out of the 41M available. We performed all experiments three times: using only monolingual score on each language, and using the bilingual score. We report only results on monolingual English method due to space; the trends were the same in all tracks.

5.1. Language Modeling

Figure 1 shows language modeling results. We present results only for monolingual English-side data selection, but the results for monolingual French-side and bilingual data selection are similar. For each of the three data selection methods, we trained language models on the most relevant subsets of various sizes. The language models were configured identically to those used for selection (order 4, and vocabulary fixed).

We evaluated these models on their perplexity on the entire TED training set (207k sentences). The recent work of [7] `"min10"` does not beat the vanilla Moore-Lewis baseline perplexity, although they converge. On the left side of Figure 1, it can be seen that proposed method of language difference models provides a clear and consistent reduction of 13 perplexity (absolute; 10% relative) over the standard word-based method. This is roughly the same perplexity improvement as was shown in [1], so adding the discriminative information to the text doubles the effectiveness of cross-entropy difference -based data selection.

The right-hand side of Figure 1 shows the number of out-of-vocabulary (OOV) tokens in the TED task corpus according to LMs trained on the selected data. We confirm the large vocabulary coverage improvement reported in [7], with `"min10"` having 43% (relative) fewer OOV's at the 2-3M selection mark. Our proposed new method is almost as good, with 37% fewer OOV's on the task.

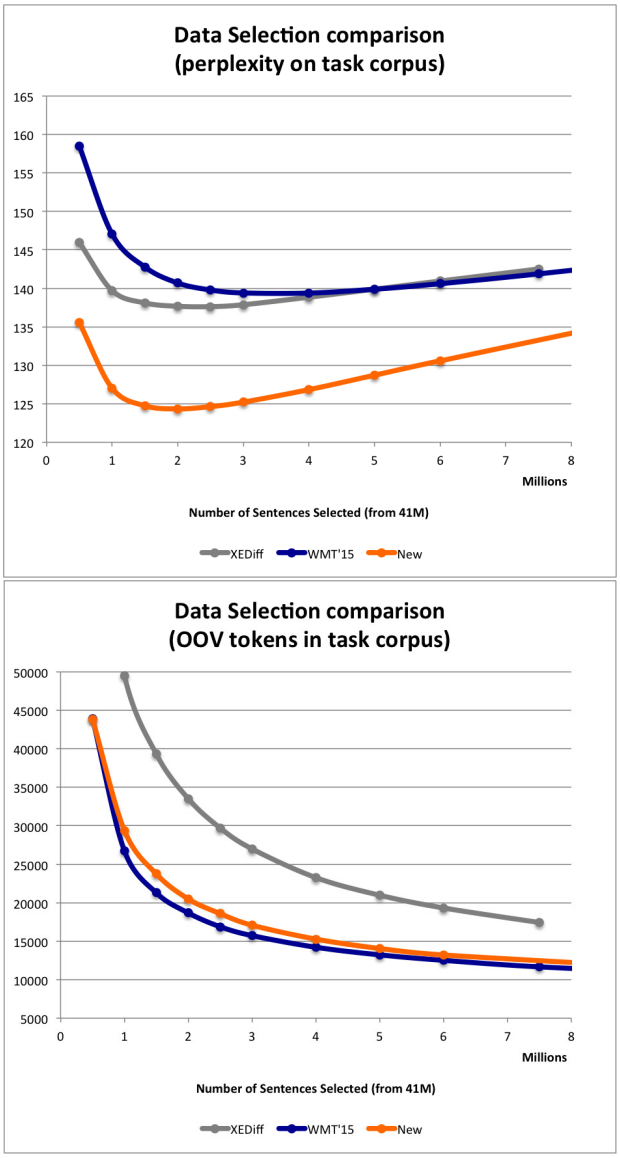


Figure 1: Comparison of perplexity scores and OOV tokens on the TED corpus for monolingual (English) data selection with word only, words-and-POS, and with language difference information.

5.2. Machine Translation

The machine translation results comparing textual representations for each data selection variant are in Figure 2. The BLEU scores of systems from the class-based language difference (`"new"`) approach are all

substantially better than either baseline on both tst2013 and tst2012.

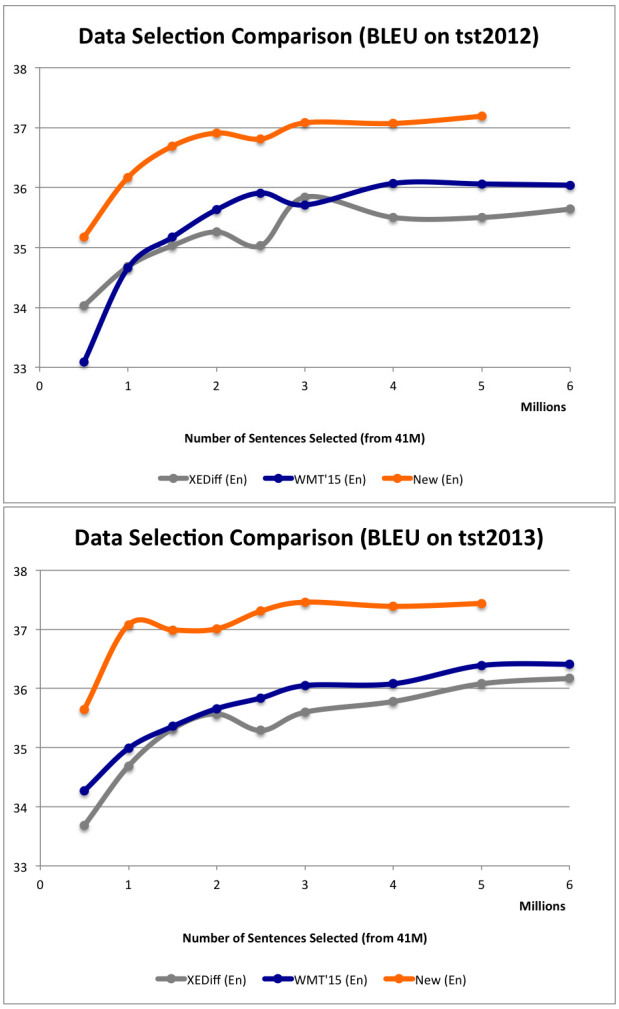


Figure 2: Comparison of BLEU scores for monolingual (English) data selection with word only, words-and-POS, and with language difference information.

When selecting 3M selected sentences and evaluating on the most recent public test set, `test2013`, the Moore-Lewis baseline of [1] has a BLEU score of 35.60, the fewer-words (”min10”) baseline from [7] scores 36.05 (+0.45), and our new method scores 37.46, +1.85 BLEU over the first baseline and +1.4 over the second, more recent, update to the state-of-the-art. The BLEU scores of the proposed method reach a higher plateau, and do so earlier. Of note is that only the language difference models select data that outperforms the in-domain corpus (the black line labeled “TED baseline” in Figure 3.

We also tested using the selected data to build a multi-model system, where the translation model trained on selected data is used in combination with one trained on the task data. Each resulting system thus had two grammars and two language models. Figure 3 contains the results of these multi-model experiments using the monolingual (English) selection method, and evaluated on “test2013”. All the data selection methods provided some benefit when used in the multi-model setup, but the proposed method using language difference models was up to +1 BLEU better than the baseline in [7] (which did not show multi-model results), up to +1.3 BLEU than the cross-entropy difference baseline, and +2 BLEU over the in-domain data alone.

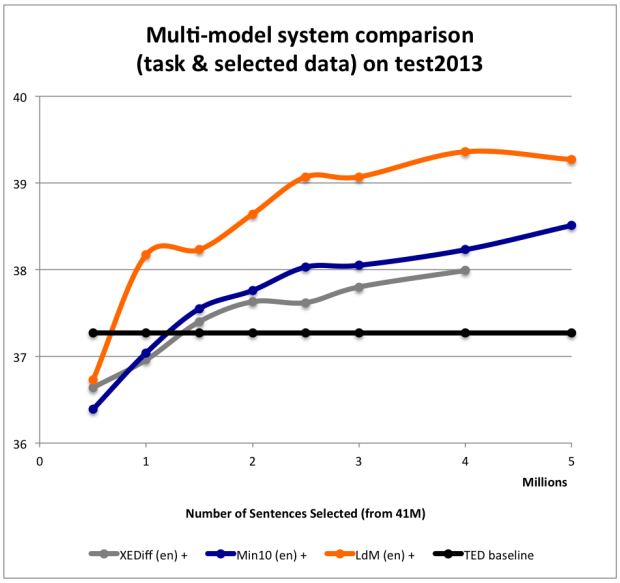


Figure 3: Using system trained on selected data as part of a two-model translation system, along with a system trained on the task corpus.

It is thus possible to use a wordless text representation to select data more usefully than a word-based method. This is surprising to us, as the language models trained on our class-based language difference text have no way of knowing if the sentences being scored are topically relevant. Modeling the difference between corpora in aggregate can thus be a stronger indicator of relevance than the words themselves for selection. We collapsed all of the words in the vocabulary as a pathological test case; a more finely-tuned approach would perhaps distinguish between words to keep and words to abstract away into a difference class.

5.3. Model Size Improvements

In addition to the translation system improvements, the memory requirements for the data selection pro-

cess itself dominated by the language model built using the data pool is dramatically smaller with our class-based n-gram language difference representation than the baseline models. The standard data selection method requires training a 12GB (binarized) language model over each side of the full 42M sentence pool in order to compute the cross-entropy score according to the general-domain corpus. The equivalent full-corpus model using our approach is 126M, or 1% as large, because the vocabulary size is negligible.

5.4. Requirements

One drawback to this use of language difference models as presented here is our use of a part-of-speech tagger in at least one of the languages. Languages with large amounts of data generally seem to have POS taggers already developed. However, there are plenty of languages for which such linguistic tools are not accessible. To construct the language difference model, the discriminative (or skew) information about each word is combined with some generalization or group label for the word that conveys part of the word's information in the sentence. POS tags are just one of many ways of grouping words together so as to capture underlying relationships within a sentence. As such, we hypothesize that other methods, such as Brown clusters [19] or topic model labels, would suffice. In the case where no word clustering method at all is available nor can be induced for the language, it seems doubtful that one could have enough data where data selection would do any good.

6. Conclusion

The data selection method of [1] directly uses the fact that the in-domain and general corpora differ in order to quantify the relevance of sentences in a data pool to an in-domain task text. This relevance is based on how much a sentence is like the in-domain corpus and unlike the pool corpus.

We have presented a way to further leverage the discriminative mechanics of the Moore-Lewis data selection process to distill a corpus down to a representation that explicitly encodes differences between the corpora for the specific data selection scenario at hand. We do this by replacing every word in the corpora with its part-of-speech tag plus a suffix that indicates the relative bias of the word, or how much likelier it is to be in the task corpus versus the pool.

Language models trained on data selected with our approach have -13 lower absolute perplexity on in-domain data than the baselines, doubling the effectiveness of the cross-entropy difference based method. The trained language models also had 37% fewer OOV's on the task data than the standard baseline. Furthermore, machine translation systems trained on data selected with our approach outperform MT systems trained on data selected with regular n-gram models by up to +1.8 BLEU, or can be stacked with in-domain translation model for up to +1.3 BLEU. These improvements come despite using zero of the original words in the texts for our selection process, and reducing the corpus vocabulary to under 200 automatically-derived tags.

By changing the representation of the text, we can use basic n-gram models to characterize the difference between the corpora. This process enables us to use common models with robust statistics that are tailored to computing the similarity score, instead of training a separate classifier or ignoring the textual differences as the standard approach does.

As a bonus, our new representation and language difference models mean that the data selection process itself is now no longer memory-bound. Because the corpus vocabulary is so compact, the language models required are also much smaller, and ordinary computational resources now suffice to perform data selection on practically any size corpus.

Much work remains, as there are surely other useful factors and more nuanced representations. What else is there about a task that differentiates its language from others, how can we quantify these features, and which of them are useful when measuring the difference between two texts? We have not explored the parameter space for our approach, either. One might wish in the future to try use powers of 2, or e, or linear bucket ranges, or adjust the ranges to ensure words are evenly distributed amongst buckets. Furthermore, one might not want to collapse the most discriminative words – the ones with the highest contribution to the cross-entropy difference score – into the same classes based on POS tag. It might be the case that it is only important to lump the least discriminative words together so as to focus the selection model on the differences between the corpora.

7. Acknowledgements

We appreciate the helpful feedback of Philip Resnik and the anonymous reviewers.

8. References

[1] R. C. Moore and W. D. Lewis, "Intelligent Selection of Language Model Training Data," *ACL (Association for Computational Linguistics)*, 2010.

[2] A. Axelrod, X. He, and J. Gao, "Domain Adaptation Via Pseudo In-Domain Data Selection," *EMNLP (Empirical Methods in Natural Language Processing)*, 2011.

[3] S. Mansour, J. Wuebker, and H. Ney, "Combining Translation and Language Model Scoring for Domain-Specific Data Filtering," *IWSLT (International Workshop on Spoken Language Translation)*, 2011.

[4] L. Wang, D. F. Wong, L. Chao, J. Xing, Y. Lu, and I. Trancoso, "Edit Distance : A New Data Selection Criterion for Domain Adaptation in SMT," *RANLP (Recent Advances in Natural Language Processing)*, no. September, pp. 727–732, 2013.

[5] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation Data Selection using Neural Language Models: Experiments in Machine Translation," *ACL (Association for Computational Linguistics)*, 2013.

[6] H. Cuong and K. Sima'an, "Latent Domain Translation Models in Mix-of-Domains Haystack," *COLING (International Conference on Computational Linguistics)*, 2014.

[7] A. Axelrod, P. Resnik, X. He, and M. Ostendorf, "Data Selection With Fewer Words," *WMT (Workshop on Statistical Machine Translation)*, 2015.

[8] J. Burrows, "Delta: A Measure of Stylistic Difference and a Guide to Likely Authorship," *Literary and Linguistic Computing*, vol. 17, no. 3, pp. 267 –287, 2002.

[9] D. Biber, *Variations Across Speech and Writing*. Cambridge, UK: Cambridge University Press, 1988.

[10] H. Baayen, H. V. Halteren, and F. Tweedie, "Outside the cave of shadows: using syntactic annotation to enhance authorship attribution," *Literary and Linguistic Computing*, vol. 11, no. 3, pp. 121–132, 1996.

[11] S. Argamon, M. Koppel, and G. Avneri, "Routing documents according to style," *Workshop on Innovative Information Systems*, vol. 60, no. 6, pp. 581–3, 1998.

[12] M. Koppel, N. Akiva, and I. Dagan, "A Corpus-Independent Feature Set for Style-Based Text Categorization," *IJCAI Workshop on Computational Approaches to Style Analysis and Synthesis*, 2003.

[13] A. Bisazza and M. Federico, "Cutting the Long Tail : Hybrid Language Models for Translation Style Adaptation," *EACL (European Association for Computational Linguistics)*, pp. 439–448, 2012.

[14] K. Papineni, S. Roukos, T. Ward, and W.-j. Zhu, "BLEU: a method for automatic evaluation of machine translation," *ACL (Association for Computational Linguistics)*, 2002.

[15] C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, P. Blumson, H. Setiawan, V. Eidelman, and P. Resnik, "cdec: A Decoder, Alignment, and Learning Framework for Finite-State and Context-Free Translation Models," *ACL (Association for Computational Linguistics) Interactive Poster and Demonstration Sessions*, 2010.

[16] D. Chiang, Y. Marton, and P. Resnik, "Online large-margin training of syntactic and structural translation features," *EMNLP (Empirical Methods in Natural Language Processing)*, 2008.

[17] K. Heafield, "KenLM : Faster and Smaller Language Model Queries," *WMT (Workshop on Statistical Machine Translation)*, 2011.

[18] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network," *NAACL (North American Association for Computational Linguistics)*, 2003.

[19] P. F. Brown, P. V. DeSouza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai, "Class-Based n-gram Models of Natural Language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

Morphology-Aware Alignments for Translation to and from a Synthetic Language

Franck Burlot, François Yvon

LIMSI, CNRS, Université Paris Saclay, 91 403 Orsay, France

firstname.lastname@limsi.fr

Abstract

Most statistical translation models rely on the unsupervized computation of word-based alignments, which both serve to identify elementary translation units and to uncover hidden translation derivations. It is widely acknowledged that such alignments can only be reliably established for languages that share a sufficiently close notion of a word. When this is not the case, the usual method is to pre-process the data so as to balance the number of tokens on both sides of the corpus. In this paper, we propose a *factored alignment model* specifically designed to handle alignments involving a synthetic language (using the case of the Czech:English language pair). We show that this model can greatly reduce the number of non-aligned words on the English side, yielding more compact translation models, with little impact on the translation quality in our testing conditions.

1. Introduction

Most statistical translation models rely on the unsupervized computation of word-based alignments, which serve both to identify elementary translation units, as in phrase-based [1] and hierarchical [2] Machine Translation (MT) and to uncover hidden translation derivations, as in n-gram-based MT [3]. The *de-facto* standard for computing such alignments is to use the IBM models [4], as implemented in efficient software packages such as GIZA++ [5, 6] or `fast_align` [7].

It is however widely acknowledged that such alignments can only be reliably established for languages that share a sufficiently close notion of a word. When this is not the case, the usual method is to pre-process the data so as to balance the number of tokens on both sides of the corpus. Assuming translation into English from a morphologically rich language, this process will decompose complex source forms into shorter segments, and/or neutralize morphological variations that are not overly marked (and thus not necessary for the translation process) in the morphologically simpler one: forms that only differ in their case marking can, for instance, be collapsed into one non-marked version for the purpose of translating into English. This situation also occurs, though in a more extreme form, when translating from a language without explicit word separators such as Chinese [8, 9].

This strategy has been successfully applied to many language pairs in the context of MT applications: [10] is a first attempt to cluster morphological variants when translating from German into English; while [11] focuses on splitting German compounds. Similar techniques have been proposed for other language pairs such as Czech [12], Arabic [13, 14], Spanish [15], Finnish [16], Turkish [17] to name a few early studies. Note that the benefits (in terms of translation quality) of such pre-processing can be limited, except for the translation of out-of-vocabulary words.

In this paper, we focus on a slightly different problem, which arises when aligning English with a synthetic language. In this situation, many English words, notably function words such as determiners, pronouns and prepositions, may have no direct equivalent on the source side, in cases where for example their function is expressed morphologically by bound morphemes. Such problems, and their detrimental consequences for MT, are more thoroughly discussed in § 2 taking the Czech:English language pair as the main source of examples. To mitigate this undesirable situation, we propose a *factored alignment model* specifically designed to handle alignments involving a synthetic language, (see § 3, where we introduce these new variants of IBM Model 2). In our experiments with MT from and into English (§ 4), we show that this model can greatly reduce the number of non-aligned words on the English side, yielding more compact translation models, with little impact on the translation quality in our testing conditions. We finally discuss related work (§ 5) and conclude with further prospects.

2. Alignments with a Synthetic Language

Czech is a morphologically rich language with complex nominal, adjectival and verbal inflection systems. For instance, compared to the English adjective, which is invariable, its Czech counterpart has many different forms, varying in case (7), number (2) and gender (3). Therefore, Czech words contain more information than in English, which is typical of a synthetic language. On the other hand, the same kind of information may be encoded in a separate word in English, a language that has analytical tendencies. For instance, the Czech nominal genitive marker plays a similar role to the English preposition *of*, as in *the engine **of** the car* $\rightarrow$ *motor aut**a***.

Therefore, when aligning those two languages, linking a Czech noun (or verb, or adjective) solely to its English counterpart is quite unsatisfactory, since the information encoded in the Czech word ending is then lost (see Figure 1);

Table 1: Unaligned preposition causing a mistake (Czech-English).

source	Na seznamu jsou **v první řadě** plány na rozsáhlejší spolupráci v oblasti jaderné energetiky.
output	On the list are the **first in a series of** plans for greater cooperation in the field of nuclear energy.
ref.	High on the agenda are plans for greater nuclear co-operation.

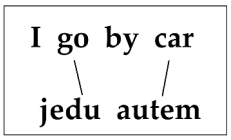


Figure 1: Lexical alignments missing the English pronoun and preposition that are encoded in the Czech endings.

and it might be desirable to also align neighboring function words on the English side. Missing these links indeed leads to mistakes in the output. In the Moses [18] baseline for Czech to English described in § 4, we often observed that an unaligned English preposition is associated to the wrong phrase, leading to a translation error, as illustrated in Table 1. In this example, the Czech *v první řadě* means literally *in first-Locative rank-Locative* and the phrases that were selected incorrectly include prepositions that were not aligned:

- **v první - first in**: this phrase pair leaves out the translation of the Czech preposition *v* and includes an English preposition that has no equivalent in the source, and might be erronously aligned to *v*.
- **řadě - a series of**: the Czech locative case is not translated and the English preposition *of* is not present on the Czech side.

We observe that standard alignment toolkits tend to miss such links. Table 2 reports the ratio of English words that remained unaligned after we trained alignments in both directions with symmetrization, using `fast_align`. Among the 7% unaligned words, almost 50% are determiners, which was predictable, since Czech does not have articles. Prepositions account for 33.2% of the unaligned words, over 10 points more than what we observe when aligning French and English. A similar situation happens with Russian, where more than 20% of English prepositions have no alignment. This suggests a difference between languages with synthetic tendencies such as Czech or Russian and more analytical ones such as English in the way they encode grammatical features such as case. When running asymmetric alignments from Czech to English, numbers are even worse, with 52.9% of the English prepositions remaining unaligned. We conclude that there is often no preposition on the Czech side to be linked to an English one. On the contrary, aligning French or Spanish to English means fewer unlinked prepositions and a higher rate of unaligned nouns. Hence, the problem of function word alignments is less obvious and the information we lose the most is lexical, rather than grammatical.

We argue that a more suitable alignment should extract phrases in which the English preposition is more systematically co-aligned with its head noun. This would make the extraction of phrases with a dangling, unaligned *of* less likely, and contribute to fixing certain case agreement errors.

Unaligned words are not only a problem in terms of the translation of prepositions. Since Czech is a pro-drop language, many English subject personal pronouns have no source to align to, leading to their omissions in many hypothesis translations when translating into English, such as in the clause with no subject found in one of the outputs of our baseline systems *and will go into it*. Aligning more English pronouns to Czech verbs should help to capture the necessity of jointly translating a verb into a pronoun and a verb in the target. In our English-to-Czech baseline (§ 4), we also often encounter situations where a negative Czech verb is translated into an affirmative form in English. Since Czech negation is encoded as a prefix (*ne-*, see Table 3), it is difficult to align it to English words such as *not*.[1]

Note that the units we need to find alignments for on the Czech side always encode grammatical information: person, negation and case, which should align to English function words. This is the main motivation for our proposal to add morphological alignments on top of lexical ones.

3. Morphological Alignment Model

3.1. Aligning words with feature vectors

Our model aims to make word-to-word alignments more dense by linking morphological tags on the Czech side to English function words. We first perform a morphological analysis of Czech and obtain a vector-based representation for each token, containing the lemma and various morphological labels (see § 2). Our model thus assumes sentences taking the form of a vector $\mathbf{e}$ of I word forms on the English side and of a $K \times J$ matrix $\mathbf{f}$ on the Czech side, where each row corresponds to various features of the word (such as lemma, person and case, as shown in Figure 2.a). By convention, we assume that the lemma is at index 1 in vector f_j.

Using these notations, our alignment model is a simple variant of IBM model 2 where (a) lemmas are aligned independently from one another, and (b) tag alignments are inde-

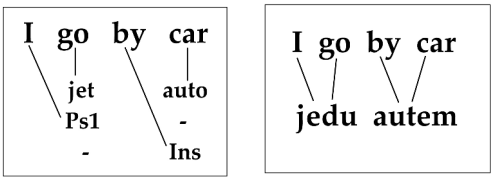
[1] The adverb *not* makes up the majority of unaligned adverbs in Table 2.

Table 2: Unaligned English words with symmetrized alignments across four language pairs using `fast_align`. $\frac{POS}{unali.}$: rate of unaligned occurrences of the POS over all unaligned words ; $\frac{unali.}{POS}$: rate of unaligned words over all occurrences of the POS.

POS	Cs-En (asym)		Cs-En (sym)		Ru-En		Fr-En		Es-En	
	$\frac{POS}{unali.}$	$\frac{unali.}{POS}$	$\frac{POS}{unali.}$	$\frac{unali.}{POS}$	$\frac{POS}{unali.}$	$\frac{unali.}{POS}$	$\frac{POS}{unali.}$	$\frac{unali.}{POS}$	$\frac{POS}{unali.}$	$\frac{unali.}{POS}$
Determiners	26.2%	65.2%	48.7%	30.1%	16.2%	31.0%	13.0%	11.6%	15.1%	4.4%
Prepositions	28.6%	52.9%	33.2%	15.3%	19.1%	23.3%	20.1%	12.4%	32.4%	7.2%
Auxiliaries	9.7%	37.6%	4.3%	4.4%	5.4%	19.5%	6.4%	11.8%	11.9%	5.6%
Nouns	8.7%	8.8%	3.4%	0.9%	26.7%	14.8%	28.6%	7.6%	8.1%	1.1%
Adverbs	4.9%	26.8%	1.9%	2.5%	3.6%	17.8%	3.2%	9.6%	6.3%	4.1%
Pers. Pronouns	7.3%	65.5%	0.6%	1.2%	2.5%	15.8%	1.6%	9.9%	3.0%	2.5%
Aligned words	72.0%		93.0%		81.6%		90.3%		96.3%	

Table 3: Unaligned negation adverb causing a mistake (English-Czech).

source	he is **not** at all aggressive
output	je vůbec agresivní
	he is at all aggressive
ref.	**ne**ní vůbec agresivní
	he is not at all aggressive



(a) (b)

Figure 2: Morphological alignments. (a) The source 1st person tag is aligned to the target pronoun *I* and the instrumental case tag to the preposition *by*. (b) Lemma and tag alignments are merged to provide links between word forms.

pendent given the alignment of their lemma, yielding:

$$p(\boldsymbol{f}|\boldsymbol{e}) = \sum_{\boldsymbol{a}} \prod_{j=1}^{J} \Big[p(a_{j1}|\boldsymbol{e}) p(f_{j1}|e_{a_{j1}}) \times \prod_{k=2}^{K} p(a_{jk}|a_{j1}) p(f_{jk}|e_{a_{jk}}) \Big] \quad (1)$$

This model thus allows us to integrate into the alignment probability the morphological properties of a lemma, which should for instance reinforce the alignment of a Czech noun with an English noun when the former is marked with a case that often matches a nearby preposition of the latter. Note that using the IBM model 2 is somewhat oversimplistic, as it assumes for instance that morphological markers of close words are unrelated, even though agreement rules enforce similar cases for words within the same noun phrase. A more realistic version, in which such dependencies would be modeled at least indirectly, would be to use a better distortion model to constrain the alignment of neighboring lemmas. Given the implementation choices described above, it was not necessary to develop this idea any further.

To complete the description, note that we assume that the alignment of the lemma (a_{j1}) only depends on j, I and J; and that the alignments of the morphological tags (a_{jk}) only depend on the difference ($a_{jk} - a_{j1}$). We further enforce $p(a_{jk}|a_{j1}) = 0$ outside of a fixed-size window centered on a_{j1} (3 words to the left side, one word to the right side).[2] The model defined in Equation (1) lends itself well to estimation via EM. We however also performed experiments with more constrained implementations, as described below.

3.2. Implementation variants

In the experiments reported below, we constrast various implementations of this alignment model in the computation of the Czech-to-English alignments; note that we use a standard word-based IBM model for the other direction. A first condition (joint//ibm in Table 10) uses a faithful implementation of EM for the model of Equation (1), in which we initialize uniformly the translation and the distortion parameters.

A second condition uses the output of a first pass alignment to better constraint the alignments of lemmas. The first stage computes alignments between Czech lemmas and English words using standard word alignment pipelines: in our experiments, we used both asymmetric alignments computed with IBM model 2 and IBM model 4, or symmetrized alignments obtained by running these models in both directions. In any case, we keep these alignment links fixed during the second stage, in which we estimate the morphological alignment model and compute alignments links for tags.

A softer version of the second condition is to use the first pass alignments to initialize the translation model, which are then free to change in the course of the EM procedure.

Finally note that we also enforce a void alignment for "null" morphological tags (eg. the case marking for verbs, or the tense of nouns, see Figure 2.b).

For all conditions, training involves multiples iterations

[2] As for the right side, we consider only one position to target words like *not* and *'s*, as in *can **not**, Hana **'s** hand*.

of EM with models of increasing complexity for a fixed number of iterations. We first train the lemma-to-word alignments, before also considering the tags-to-word parameters. A final decoding computes the optimal alignment for morphological tags; at this stage, we only keep alignment links that match a non-aligned word on the English side, and use these to complete the baseline alignment, as shown in Figure 2.b. The rest of the training of the translation model (phrase extraction, etc.) remains unchanged.

4. Experimental Results

4.1. Data and Experimental Setup

We used two datasets to train our SMT systems:

- **A small dataset** consisting of about 790k parallel sentences taken from the Europarl [19] and News Commentary corpora distributed for the shared translation task of WMT 2015.[3] The monolingual data is made up of one side of the parallel corpora and the News Crawl corpora (2014) and adds up to 29M sentences for English and 37M for Czech.

- **A bigger dataset** of about 15M parallel sentences, composed of the previous set and the Czeng 1.0 corpus [20]. We added to the monolingual data one side of the Czeng 1.0 corpus and the previous versions of the News Crawl corpora (2007-2013). and obtained a total of 52M Czech and 43M English sentences.

This data is tokenized and true-cased before starting the alignment. The morphological analysis on the Czech side is performed using MorphoDiTa [21]. After word alignment, all downstream training steps are carried out using the Moses toolkit [18]: this includes phrase extraction and scoring, lexical weighting and learning the lexicalized reordering models. 4-gram language models are trained with KenLM [22] for both languages. Tuning is performed using MERT [23] on the test set of the WMT 2014 translation task. For the sake of comparison, we also report results obtained with n-gram-based systems trained with Ncode [3, 24].

4.2. Alignment Setup

We used M morphological features to fill the Czech word vectors $\mathbf{f}$ in our experiments: case, person, time/mode, and whether a verb has a negative form - Czech representations have therefore $M = 5$ dimensions.

Regarding condition 1, where lexical alignments are learnt jointly with morphological links (for Czech-to-English), 4 strategies were tested:

- **ibm//none**: only forward (cs-en) alignments;

- **joint//none**: only forward (cs-en) alignments trained according to our model;

- **ibm//ibm**: forward and backward alignments symmetrized with the grow-diag-final-and heuristics;

- **joint//ibm**: symmetrization is performed with joint-none and the backward (en-cs) alignments;

Regarding the training condition 2, we used `fast_align` (resp. `Mgiza`) to get initial IBM2 (resp. IBM4) alignments between Czech lemmas and English words. We added to the former 3 strategies to obtain different alignment types:

- **ibm+morph//none**: forward and morphological alignments;

- **ibm+morph//ibm**: a symmetrized version also involving backward en:cz alignments;

- **[ibm//ibm]+morph**: morphological alignment is performed after symmetrization.

During decoding, the most likely morphological alignments are subject to three constraints in order to be accepted:

- The candidate English lemma should not be aligned;

- The morphological alignment probability should be higher than a threshold (0.05 in our experiments);

- The candidate English lemma should have a frequency higher than 1,000 occurrences (15,000 for the bigger data set) in the English part of the parallel corpus.

These heuristics help to improve the quality of alignment by reducing links with rare words that may have a high probability, given a specific tag. Since the words we target are mainly English function words (pronouns, prepositions, etc.), it seems reasonable to focus on a small set of high frequency tokens. Note finally that the same word alignments were used both to train the en-cs and the cs-en systems.

4.3. Results

Morphological alignments effectively address the problem of previously unaligned words by linking function words, as reflected in Table 4, even though ibm+morph//none also returns a few more alignments for nouns. This shows that some lexical alignments had also been wrongly performed, most of which are corrected by symmetrization in the ibm+morph//ibm variant. The first impact of morphological alignments is a reduction of the phrase table size: using `fast_align`, we lost almost 1.5M phrases when adding morphological alignments to the symmetrized baseline, meaning that over 6% of initial phrases have been discarded (see Table 5).[4] `Mgiza` alignements show the clearest contrast, since the number of phrase pairs for ibm//ibm (44M) is reduced to less than 28M in ibm+morph//ibm.

[3] `http://statmt.org/wmt15/`

[4] Note that if the number of phrase pairs is lower, the average length of phrases stay the same in every system. For instance, ibm//ibm has 3.77 tokens per Czech phrase and 4.26 per English one, which is very similar to [ibm//ibm]+morph with respectively 3.79 and 4.25 tokens per phrase.

Table 4: Links added by morphological alignments (Czech-English) using `fast_align`. $\frac{POS}{unali.}$: rate of unaligned occurrences of the POS over all unaligned words ; $\frac{unali.}{POS}$: rate of unaligned words over all occurrences of the POS.

POS	ibm//none		ibm+morph//none		ibm//ibm		[ibm//ibm]+morph		joint//ibm	
	$\frac{POS}{unali.}$	$\frac{unali.}{POS}$	$\frac{POS}{unali.}$	$\frac{unali.}{POS}$	$\frac{POS}{unali.}$	$\frac{unali.}{POS}$	$\frac{POS}{unali.}$	$\frac{unali.}{POS}$	$\frac{POS}{unali.}$	$\frac{unali.}{POS}$
Determiners	26.2%	65.2%	32.6%	58.2%	48.7%	30.1%	58.7%	28.5%	51.6%	24.3%
Prepositions	28.6%	52.9%	25.6%	34.0%	33.2%	15.3%	24.4%	8.8%	31.3%	11.0%
Auxiliaries	9.7%	37.6%	7.0%	20.6%	4.3%	4.4%	3.3%	2.7%	4.5%	3.5%
Nouns	8.7%	8.8%	9.4%	6.9%	3.4%	0.9%	3.4%	0.7%	3.0%	0.6%
Adverbs	4.9%	26.8%	5.0%	19.8%	1.9%	2.5%	2.0%	2.2%	1.9%	2.0%
Pers. Pronouns	7.3%	65.5%	4.7%	25.7%	0.6%	1.2%	0.7%	1.0%	0.7%	1.0%
Aligned words	72.0%		79.3%		93.0%		94.4%		94.6%	

Table 5: Results in BLEU for Czech-English (smaller data condition).

	`fast_align` (IBM2)			`Mgiza` (IBM4)	
Alignment Setup	Ncode	Moses	Phrase Table Size	Moses	Phrase Table Size
ibm//none	-	20.34	50,462,274	20.31	56,967,921
ibm+morph//none	-	19.98	35,364,892	20.26	45,549,682
ibm+morph//ibm	-	20.08	20,286,841	20.14	27,820,416
ibm//ibm	19.72	20.34	22,799,794	20.35	44,410,638
[ibm//ibm]+morph	19.68	20.26	21,247,701	20.33	40,805,062

Table 6: Results in BLEU for English-Czech (for the small data condition). The size of the phrase tables is the same as in Table 5.

	`fast_align`		`Mgiza`
Alignment Setup	Ncode	Moses	Moses
ibm//none	-	13.94	14.24
ibm+morph//none	-	13.90	14.03
ibm+morph//ibm	-	14.02	13.91
ibm//ibm	14.02	14.09	14.45
[ibm//ibm]+morph	14.03	14.21	14.20

Table 7: Better case prediction (English-Czech).

source	who are captured by Ukrainian soldiers
ibm//ibm	kteří zadržený ukrajinští vojáci
	who-Plur captured-Passive-Sing Ukrainian-Nom soldiers-Nom
[ibm//ibm]+morph	kteří jsou zajati ukrajinskými vojáky
	who-Plur are captured-Passive-Plur Ukrainian-Ins soldiers-Ins

We evaluated our systems using the test set of the WMT 2015 translation shared task. Even though the effect on the BLEU score is minor, we observe a slight improvement when translating into Czech with `fast_align`[5] (see Table 6), which is understandable, since case is the major morphological category ignored by baseline alignments. Thus the new phrase table helps to better predict case inflection, mainly according to the preposition in the source sentence. Indeed, Table 7 shows the wrong translation of the English preposition *by* in the ibm//ibm system where the noun phrase is in nominative case. Our [ibm//ibm]+morph system successfully translates the preposition by the instrumental case needed for such passive constructions. Moreover, in the same direction, handling negation also helped to fix some baseline system errors, as for the example in Table 3 (our system actually outputs the reference sentence).

Note that a better management of case is also beneficial in the inverse direction (Czech-English), as shown in Table 8, where the erroneous phrase pairs described in § 2 (*v první - first in*;) *řadě - a series of* get a lower probability, allowing the correct translation to be selected during decoding. As a result, we observe that the most frequent prepositions (*of, to, in, for*) are generated less often in [ibm//ibm]+morph (4,070) than in the ibm//ibm (4,190), which we interpret as a sign of more relevant use of English prepositions in a morphology-aware system.

For the same translation direction, the number of subject personal pronouns is higher in [ibm//ibm]+morph (1,629) than in ibm//ibm (1,561), which suggests better constructions in the English output, such as in Table 9, where the Czech verb with no subject expressed is translated by a verb with its subject pronoun corresponding to the source word ending.

Furthermore, handling negation during the alignment step also yields improvement when translating into English. Indeed, the word *not* has 206 occurrences in ibm//ibm and 234 in [ibm//ibm]+morph, suggesting that the latter system

[5]The descriptions of our outputs relate to the alignments performed using `fast_align`.

Table 8: Better preposition extraction for relevant phrases (Czech-English).

source	Na seznamu jsou **v první řadě** plány na rozsáhlejší spolupráci v oblasti jaderné energetiky.
ibm//ibm	On the list are the **first in a series of** plans for greater cooperation in the field of nuclear energy.
[ibm//ibm]+morph	On the list are **primarily** plans for greater cooperation in the field of nuclear energy .

Table 9: Subject personal pronoun generation (Czech-English).

source	a bude**me** si ho rozebírat and will-Ps1-Plur it analyse
ibm//ibm	and will go into it
[ibm//ibm]+morph	and **we** will discuss it

conveys negation more.

Alignments with the time and mode tags for verbs helped to generate more correct English analytical constructions: while ibm//ibm omits the auxiliary in the translation of a Czech present verb into a passive form (*who usually based*), [ibm//ibm]+morph generates the right construction, despite the insertion of an adverb between both verbs: *who **are** usually based*. Nevertheless, for 2,639 auxiliaries in the former, the latter contains 2,716 of them, bringing almost insignificant changes.

We notice slightly worse results with the condition 1, where joint//ibm is 1 BLEU point below ibm//ibm for Czech-English, and 0.6 for English-Czech (see Table 10). The number of phrase pairs is a lot lower here than with condition 2, since more alignments are generated, as is shown in Table 4. Nevertheless, the score of the joint//none systems in both directions show that these alignments are very noisy, since they greatly underperform the ibm//none system.

Finally, Table 11 suggests that no impact on the BLEU score compared to the baseline is to be expected using more data, while the total ratio of aligned words went from 91.7% to 93.6% and 7% of initial phrases were discarded from the table in [ibm//ibm]+morph.

Table 10: Results in BLEU with joint learning of morphological and lexical alignments using Moses for the small data condition (+`fast_align` init: parameter initialization with `fast_align` output)

Alignment Setup	cs-en	en-cs	Phrase Table Size
ibm//none	20.34	13.94	50,462,274
joint//none	18.69	13.05	31,482,262
ibm//ibm	20.34	14.09	22,799,794
joint//ibm	19.33	13.47	15,179,849
+ `fast_align` init	19.41	13.40	15,210,792

Table 11: Results in BLEU for the large data condition (`Mgiza` with Moses)

Alignment Setup	cs-en	en-cs	Phrase Table Size
ibm//ibm	24.04	16.48	324,969,903
[ibm//ibm]+morph	24.07	16.38	301,714,878

5. Related Work

Aligning English with "morphologically-complex" languages poses several challenges, depending on the exact differences between the source and target – it has, over the years, attracted a considerable amount of effort, which has only been briefly reviewed here. In fact, morphological complexity can have multiple consequences for alignment.

First, it is often assumed that the morphologically complex language has more word types, due for instance to a richer inflectional system: this is the case for French or Spanish, which have a much richer conjugation than English. This, in turn, yields sparser counts, and less reliable probability estimates for the alignment models (notwithstanding a high Out-of-Vocabulary (OOV) ratio at testing time). The simplest remedy is to normalize the target side, using lemmas or other kinds of abstraction instead of words for the purpose of the alignment [25, 26, 27]. Note that defining the optimal level of abstraction is not obvious and often requires a significant tuning effort. Going one step further, it may also be interesting to keep these abstract representations for translation, but this requires a non-trivial post-processing step to restore the correct inflection when translating *into* the morphologically rich language [28]. The alternative strategy, which translates word forms, is plagued with OOV issues and requires specific strategies to properly handle unknown forms - as in the factored-models approach of [29, 30]. In our own alignment model, we borrow the idea to compute a first-pass alignment based primarily on lemmas, which seems to be more effective than using full forms. However, in our case, morphological information is not used to smooth alignment counts, but rather to take account of the function words in the English side.

The other well documented issue with morphologically rich languages is that word forms are more complex, meaning that they are made of several parts (morphemes for basic lexical units, lexemes for compounds). Depending on the language under consideration, identifying the orthographical and/or phonological counterparts of this elementary units can be fairly easy (in the case of purely agglutinative languages) or near impossible (in the case of fusional languages), with a large number of in-between situations. Many rule-based attempts at performing such decompositions as a pre-processing of the source side text have nonetheless been entertained: see [12], Arabic [13, 14], Spanish [15], Finnish [16], Turkish [17] to cite a few. Note that the opposite approach, consisting of "splicing" English words into artificially complex forms has also been considered (eg. in [31]).

As noted by several authors, decomposing word forms into morphemes goes against the main intuition of phrase-based SMT, which favors the translation of large units, and it also reduces the effectiveness of language models, as it decreases the size of the context. To mitigate these potentially negative effects, it is possible to simultaneously consider multiple decomposition schemes, which are then recombined using system combination techniques [32, 33, 34]. This however requires mechanisms to generate multiple morphological decompositions of the same text, using for instance the unsupervized segmentation models of [35, 36, 37]. As pointed out in [38], performing morphological segmentation of the source independently of the target is vastly sub-optimal, and joint models for alignment and segmentations are probably more appropriate in a MT context eg. [38, 39]. Our main focus being a fusional language, we have not made any attempt to segment the source words into smaller morphemes, and have instead used a feature-based representation associating a lemma and morphological properties.

6. Conclusions

This paper has described a factored alignment model specifically designed to handle alignments involving a language with synthetic tendencies, such as Czech. We have shown that this model can greatly reduce the number of non-aligned words on the English side, yielding more compact translation models that contain more relevant phrases. Case is the morphological feature that produces most alignments, which turned out to give some improvement when translating into Czech. On the other hand, using time and mode did not bring the expected gain, although it did help to better translate verb inflection in Czech and constructions in English.

The reported improvement over the baseline systems is not confirmed by a straight BLEU improvement. However we showed that one-to-many alignments from Czech to English help to better take into account the specificities of each language. While the English output has more words than in the baseline system, such as negative adverbs, auxiliaries, pronouns (disregarding the fact that it has fewer prepositions), the Czech output is more concise, showing eg. fewer incorrect verbal constructions and more reliance on inflection, which leads to better agreement.

In future work, we intend to confirm these tendencies by (a) using an improved model of morphological alignments, with an improved modeling of the dependency between tags and lemmas, and (b) testing our model with other translation tasks involving a synthetic target language.

7. Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments and suggestions. This work has been partly funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21).

8. References

[1] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. HLT–NAACL*, 2003, pp. 127–133.

[2] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," in *Proc. ACL*, Ann Arbor, MI, 2005, pp. 263–270.

[3] 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.

[4] P. F. Brown, J. Cocke, S. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to machine translation," vol. 16, no. 2, pp. 79–85, 1990.

[5] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[6] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Proc. SETQA–NLP*, 2008, pp. 49–57.

[7] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," in *Proc. NAACL*, Atlanta, Georgia, 2013, pp. 644–648.

[8] P.-C. Chang, M. Galley, and C. D. Manning, "Optimizing Chinese word segmentation for machine translation performance," in *Proc. WMT*, Columbus, Ohio, 2008, pp. 224–232.

[9] T. Chung and D. Gildea, "Unsupervised tokenization for machine translation," in *Proc. EMNLP*, Singapore, 2009, pp. 718–726.

[10] S. Nießen and H. Ney, "Toward hierarchical models for statistical machine translation of inflected languages," in *Proc. of the ACL 2001 Workshop on Data-Driven Methods in MT*, Toulouse, France, 2001, pp. 47–51.

[11] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proc. EACL*, Budapest, Hungary, 2003, pp. 187–193.

[12] S. Goldwater and D. McClosky, "Improving statistical MT through morphological analysis," in *Proc. HLT–EMNLP*, Vancouver, Canada, 2005, pp. 676–683.

[13] Y.-S. Lee, "Morphological analysis for statistical machine translation," in *Proc. HLT-NAACL 2004: Short Papers*, 2004, pp. 57–60.

[14] F. Sadat and N. Habash, "Combination of arabic preprocessing schemes for statistical machine translation," in *Proc. COLING/ACL*, 2006, pp. 1–8.

[15] A. de Gispert and J. B. Mariño, "On the impact of morphology in English to Spanish statistical MT," *Speech Communication*, vol. 50, no. 11-12, pp. 1034–1046, 2008.

[16] S. Virpioja, J. J. Väyrynen, M. Creutz, and M. Sadeniemi, "Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner," in *Proc. MT Summit XI*, Copenhagen, Denmark, 2007, pp. 491–498.

[17] K. Oflazer and I. D. El-Kahlout, "Exploring different representational units in English-to-Turkish statistical machine translation," in *Proc. WMT*, 2007, pp. 25–32.

[18] 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. ACL:Systems Demos*, Prague, Czech Republic, 2007.

[19] P. Koehn, "A parallel corpus for statistical machine translation," in *Proc. MT-Summit*, Phuket, Thailand, 2005.

[20] O. Bojar, Z. Žabokrtský, O. Dušek, P. Galuščáková, M. Majliš, D. Mareček, J. Maršík, M. Novák, M. Popel, and A. Tamchyna, "The Joy of Parallelism with CzEng 1.0," in *Proc. LREC2012*, ELRA. Istanbul, Turkey: ELRA, 2012.

[21] J. Straková, M. Straka, and J. Hajič, "Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition," in *Proc. ACL: System Demos*, Baltimore, Maryland, 2014, pp. 13–18.

[22] K. Heafield, "KenLM: Faster and Smaller Language Model Queries," in *Proc. WMT*, Edinburgh, Scotland, 2011, pp. 187–197.

[23] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. ACL*, 2003, pp. 160–167.

[24] 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.

[25] H. Ney and M. Popovic, "Improving word alignment quality using morpho-syntactic information," in *Proc. COLING*, Geneva, Switzerland, 2004, pp. 310–314.

[26] A. de Gispert, D. Gupta, M. Popović, P. Lambert, J. Mariño, M. Federico, H. Ney, and R. Banchs, "Improving statistical word alignments with morpho-syntactic transformations," in *Advances in Natural Language Processing*, T. Salakoski, F. Ginter, S. Pyysalo, and T. Pahikkala, Eds. Springer Berlin Heidelberg, 2006, vol. 4139, pp. 368–379.

[27] M. Carpuat, "Toward using morphology in French-English phrase-based SMT," in *Proc. WMT*, Athens, Greece, 2009, pp. 150–154.

[28] A. Fraser, M. Weller, A. Cahill, and F. Cap, "Modeling inflection and word-formation in SMT," in *Proc. EACL*, Avignon, France, 2012, pp. 664–674.

[29] P. Koehn and H. Hoang, "Factored translation models," in *Proc. EMNLP–CoNLL*, Prague, Czech Republic, 2007, pp. 868–876.

[30] O. Bojar, "English-to-Czech factored machine translation," in *Proc. of the 2nd WMT*, Prague, Czech Republic, 2007, pp. 232–239.

[31] N. Ueffing and H. Ney, "Using POS information for statistical machine translation into morphologically rich languages," in *Proc. EACL*, Budapest, Hungary, 2003, pp. 347–354.

[32] C. J. Dyer, "The "noisier channel": Translation from morphologically complex languages," in *Proc. WMT*, Prague, Czech Republic, 2007, pp. 207–211.

[33] A. de Gispert, S. Virpioja, M. Kurimo, and W. Byrne, "Minimum Bayes Risk Combination of Translation Hypotheses from Alternative Morphological Decompositions," in *Proc. NAACL–HLT*, Boulder, Colorado, 2009, pp. 73–76.

[34] S. Virpioja, J. Väyrynen, A. Mansikkaniemi, and M. Kurimo, "Applying morphological decompositions to statistical machine translation," in *Proc. WMT and MetricsMATR*, Uppsala, Sweden, 2010, pp. 195–200.

[35] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Trans. Speech Lang. Process.*, vol. 4, no. 1, pp. 3:1–3:34, Feb. 2007.

[36] S. Goldwater, T. L. Griffiths, and M. Johnson, "A Bayesian framework for word segmentation: Exploring the effects of context," *Cognition*, vol. 112, no. 1, pp. 21–54, 2009.

[37] D. Mochihashi, T. Yamada, and N. Ueda, "Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling," in *Proc. ACL/IJCNLP*, 2009, pp. 100–108.

[38] T. Nguyen, S. Vogel, and N. A. Smith, "Nonparametric word segmentation for machine translation," in *Proc. COLING*, Beijing, China, 2010, pp. 815–823.

[39] J. Naradowsky and K. Toutanova, "Unsupervised Bilingual Morpheme Segmentation and Alignment with Context-rich Hidden Semi-Markov Models," in *Proc. ACL*, Portland, OR, 2011, pp. 895–904.

An Open Source Toolkit for Word-level Confidence Estimation in Machine Translation

Christophe Servan[†]*, Ngoc Tien Le*[†]*, Ngoc Quang Luong*[‡]*, Benjamin Lecouteux*[†] *and Laurent Besacier*[†]

[†]GETALP – LIG, University of Grenoble Alpes, France
firstname.lastname@imag.fr
[‡]Idiap Research Institute, 1920 Martigny, Switzerland
nluong@idiap.ch

Abstract

Recently, a growing need of Confidence Estimation (CE) for Statistical Machine Translation (SMT) systems in Computer Aided Translation (CAT), was observed. However, most of the CE toolkits are optimized for a single target language (mainly English) and, as far as we know, none of them are dedicated to this specific task and freely available.

This paper presents an open-source toolkit for predicting the quality of words of a SMT output, whose novel contributions are (i) support for various target languages, (ii) handle a number of features of different types (system-based, lexical, syntactic and semantic). In addition, the toolkit also integrates a wide variety of Natural Language Processing or Machine Learning tools to pre-process data, extract features and estimate confidence at word-level. Features for Word-level Confidence Estimation (WCE) can be easily added / removed using a configuration file.

We validate the toolkit by experimenting in the WCE evaluation framework of WMT shared task with two language pairs: French-English and English-Spanish. The toolkit is made available to the research community with ready-made scripts to launch full experiments on these language pairs, while achieving state-of-the-art and reproducible performances.

1. Introduction

Statistical Machine Translation (SMT) has proven its efficiency during the last decade. For Computer Aided Translation (CAT) of documents, the following process is now broadly used: the SMT system produces raw translations then trained professional translators post-edit (correct) translation errors (PE). We believe that this SMT+PE pipeline can be improved using automatic confidence estimation (CE) where the system gives some clues about the quality of the SMT output. For instance, post-editors require to have information about the possible quality of the translation (Should they just post-edit the translation or rewrite the whole output? What are the main words/phrases they need to focus on?).

Building a method that could point out both correct and incorrect parts in SMT output is a key component to solve the above problems. When we limit the concept "parts" to "words", the automatic confidence estimation process is called Word-level Confidence Estimation (WCE).

Past years have seen the emergence of shared tasks to estimate the translation quality (like WMT CE shared task[1]). In 2015, the organizers called for methods to predict the quality of SMT output at run-time on three levels: sentence-level (Task 1), word-level (Task 2) and (new) document-level (Task 3). This paper more precisely deals with the second task (WCE) but we believe it might be of interest to researchers who work in quality assessment for SMT.

Contributions Our experience in participating in *task 2* (WCE) leads us to the following observation: while feature processing is very important to achieve good performance, it requires to call a set of heterogeneous Natural Language Processing tools (for lexical, syntactic, semantic analyses). Thus, we propose to unify the feature processing, together with the call of machine learning algorithms, to facilitate the design of confidence estimation systems. The open-source toolkit proposed (written in *Python* and made available on *GitHub*) integrates some standard as well as in-house features that have proven useful for WCE (based on our experience in WMT 2013 and 2014).

Outline The paper is organized as follow: Section 2 presents WCE task and related works on this topic. Section 3 is an overview of the features we extract while Section 4 describes the toolkit itself. Performances obtained using our WCE toolkit are given in Section 5 while Section 6 illustrates how one can easily apply feature selection for WCE using the provided code. Finally, Section 7 concludes this work and gives some perspectives.

2. WCE formalisation and related work

2.1. WCE formalisation

Machine translation (MT) consists in finding the most probable target language sequence $\hat{e} = (e_1, e_2, ..., e_N)$ given a source language sentence $f = (f_1, f_2, ..., f_M)$. We can represent Word-level Confidence Estimation (WCE) information as a sequence q (same length N of $\hat{e}$) where $q =$

[1]Since 2012 (http://www.statmt.org/wmt12/quality-estimation-task.html)

$(q_1, q_2, ..., q_N)$ and $q_i \in \{good, bad\}$[2]. Basically, the WCE component solves the equation[3]:

$$\hat{q} = \underset{q}{\operatorname{argmax}}\{p(q|f, e)\} \quad (1)$$

This is a sequence labelling task that can be solved with several Machine Learning techniques such as Conditional Random Fields (CRF) [1]. However, to train sequence labelling models, we need a large amount of training data for which a triplet (f, e, q) is available. In our case, we use binary labels associated to each word: *Good* or *Bad* to indicate whether a word is "correct" or "incorrect", respectively.

2.2. Related work

According to [2], features for Word-level Confidence Estimation (WCE) can be classified in two types regarding their origin: the "external features" and the "internal features". On the one hand, internal features are extracted from the SMT system itself like alignment table, N-best list, word graph, *etc.* On the other hand, external features mainly come from linguistic knowledge sources like syntactic parser, WordNet or BabelNet API, *etc.* In our approach, we use both types of features. They are mostly detailed in Section 3.

The first works about confidence estimation [3, 4], focused at the word level, was inspired by work done in automatic speech recognition [5]. The combination of a large amount of features, through a Naive Bayes model and a Neural Network, showed that Word Posterior Probability (WPP) was the most relevant internal feature. Later on, [6] integrated POS tagging and other external features. In the same way, [7] proposed 70 linguistic features for quality estimation at sentence level. Some of these features can be applied at word level. Their work also revealed the need of efficient machine learning algorithms to integrate multiple features and achieve better performance.

Recent workshops proposed some shared evaluation tasks of WCE systems, in which several attempts of participants to mix internal and external features were successful. The estimation of the confidence score uses mainly classifiers like Conditional Random Fields [8, 9], Support Vector Machines [10] or Perceptron [11].

Further, some investigations were conducted to determine which feature seems to be the most relevant. [10] proposed to filter features using a forward-backward algorithm to discard linearly correlated features. Using Boosting as learning algorithm, [2] was able to take advantage of the most significant features.

Our work, inspired by all those previous papers, proposes to mix internal and external features and uses CRF as decision algorithm to estimate a WCE score. The technical novelty is their integration in a single toolkit, with ready-made scripts, to quickly run reproducible experiments on different language pairs. It also provides a built-in feature selection approach. Contrarily to the toolkit proposed in [12], our framework allows a quick and easy reproduction of the results presented in this paper and addition of new features is straightforward.

[2] q_i could be also more than 2 labels, or even scores but this paper only deals with error detection (binary set of labels).

[3] In the equation, p is a probability but it could be any scoring function.

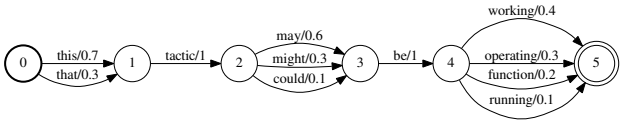


Figure 1: Example of Confusion Network

3. Available Features

Our toolkit extracts several internal and external features to train a classifier, as indicated in Table 1. These features were chosen because of their relevance in previous Word-level Confidence Estimation tasks [13, 14, 15]. Some of them are already described in detail in some previous papers [5, 3, 4, 6, 10, 2, 16]. Consequently, the novel features, which we added into our current toolkit, are in **"bold"** in Table 1. Also, the features in *"italic"* are conventional features but extracted using a new approach.

The feature list could be extended (by us or by other contributors) in the future, since the toolkit is made available to the research community. For instance, we plan to integrate the use of monolingual or bilingual word embeddings following the works of [17].

It is important to note that our toolkit extracts the features regarding *tokens* in the machine translation (MT) hypothesis sentence. In other words, one feature is extracted for each token in the MT output. So, in the Table 1, *target* refers to the feature coming from the MT hypothesis and *source* refers to a feature extracted from the source word aligned to the considered target word. More details on some of these features are given in the next subsections.

3.1. Internal Features

These features are given by the Machine Translation system, which outputs additional data like N-best list.

In addition to features corresponding to source / target words or POS (feat. 5 to 10), **Word Posterior Probability** (WPP), **WPP Max**, **WPP Min** and **Nodes** features are extracted from a confusion network, which comes from the output of the machine translation N-best list. **WPP Exact** is the WPP value for each word concerned at the exact same position in the graph. **WPP Any** extract the same information at any position in the graph. **WPP Min** gives the smallest WPP value concerned by the transition and **WPP Max** its maximum.

In the example shown in Figure 1, the target word *"function"* gets a **WPP Exact** at *0.2*, **WPP Min** at *0.1* and **WPP max** at *0.4*.

1 *Proper Name*	9 Target Word	17 WPP Any*	25 *Constituent Label*
2 **Unknown Stem**	10 **Target Stem**	18 WPP Min*	26 *Distance To Root*
3 # of Word Occurrences	11 *Word context Alignements*	19 WPP Max*	27 *Polysemy Count – Target*
4 **# of Stem Occurrences**	12 *POS context Alignements*	20 Nodes	28 **Occur in Bing Translator**
5 Source POS	13 **Stem context Alignements**	21 Numerical	29 Occur in Google Translate
6 *Source Word*	14 Longest Target N-gram Length	22 Punctuation	
7 **Source Stem**	15 Longest Source N-gram Length	23 Stop Word	
8 Target POS	16 WPP Exact*	24 Target Backoff Behaviour	

Table 1: Features extracted by the toolkit: highlights in **"bold"** are the new features we propose, the other features are those classically extracted ; we put in *"italic"* those for which we propose a new extraction method compared to previous work (see Section 4.2.3). Features indicated with " * " are internal ones.

3.2. External Features

Below is the list of the external features we use in our toolkit:

- **Proper Name**: indicates if a word is a proper name (same binary features are extracted to know if a token is **Numerical**, **Punctuation** or **Stop Word**).
- **Unknown Stem**: informs whether the stem of the considered word is known or not.
- **Number of Word/Stem Occurrences**: count the occurrences of a word/stem in the sentence.
- **Alignment context features**: these features (#11-13 in Table 1) are based on collocations and proposed by [18]. Collocations could be an indicator for judging if a target word is generated by a particular source word. We also apply the reverse, the collocations regarding the source side:
  - *Source alignment context features*: the combinations of the target word, the source word (with which it is aligned), and one source word before and one source word after (left and right contexts respectively).
  - *Target alignment context features*: the combinations of the source word, the target word (with which it is aligned), and one target word before and one target word after.

  With the example presented in Table 2, the target word "of" is aligned with "de". The source context extracted corresponds to the two words around "de", which are "nature" and "l' ". The *source alignment context features* are "of/nature", "of/de" and "of/l' " In the same way, he *target alignment context features* of "de" are: "de/nature", "de/of" and "de/the".

  We applied the same context extraction for Part-of-Speech and Stems.

Target	the	nature	of	the	independence	granted	...
Source	la	nature	de	l'	indépendance	octroyée	...

Table 2: Example of parallel sentence where words are aligned one-to-one.

- **Longest Target (or Source) N-gram Length**: we seek to get the length $(n+1)$ of the longest left sequence (w_{i-n}) concerned by the current word (w_i) and known by the language model (LM) concerned (source and target sides). For example, if the longest left sequence w_{i-2}, w_{i-1}, w_i appears in the target LM, the longest target n-gram value for w_i will be 3. This value ranges from 0 to the max order of the LM concerned.

- The word's constituent label (**Constituent Label**) and its depth in the constituent tree (**Distance to Root**) are extracted using a syntactic parser, the Figure 2 illustrates the distance between a word and its root in the tree. In the case of *"working"*, the **Constituent Label** is *VBG* and the **Distance to Root** value is *6*.

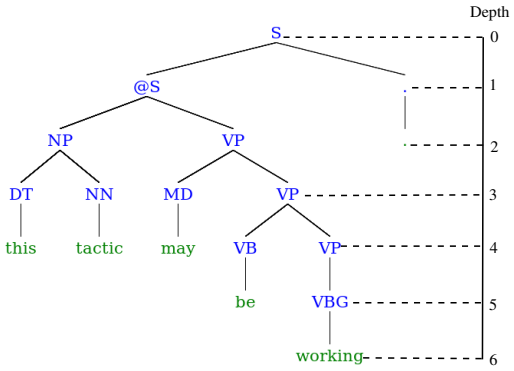


Figure 2: Example of constituent tree.

- **Target Polysemy Count**: we extract the polysemy count, which is the number of meanings of a word in a given language.

- **Occurences in Google Translate** and **Occurences in Bing Translator**: in the translation hypothesis, we (optionnally) test the presence of the target word in online translations given respectively by *Google Translate* and *Bing Translator*.

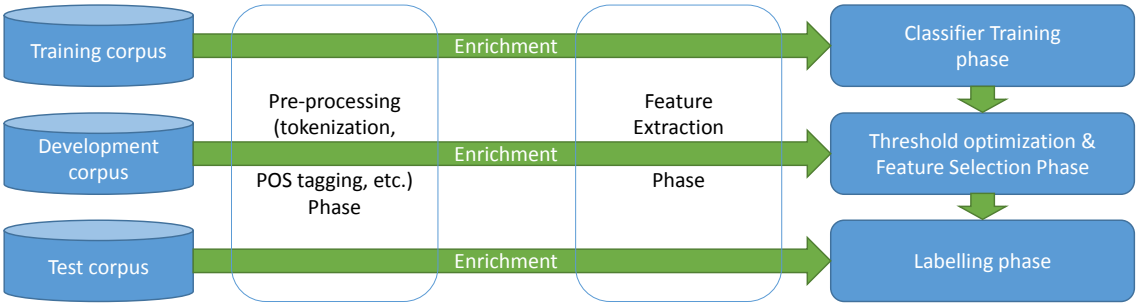


Figure 3: Pipeline of our Word-level Confidence Estimation tool

4. Toolkit

In this section, we detail our toolkit, which is a complete out-of-the-box Word-level Confidence Estimation (WCE) system. It is a customizable, flexible, and portable platform.

4.1. Pipeline Overview

Our toolkit is described in Figure 3. It contains three essential components: *preprocessing*, *feature extraction* and *training / labeling*. It integrates several existing Natural Language Processing (NLP) tools and API. It is developed in *Python 3* to use efficiently existing libraries/toolkits as well as being object-oriented designed.

The source code is available on a *GitHub* repository[4] and provided with ready-made scripts to run reproducible experiments on a French–English WCE task (for which the data is also made available).

4.2. System Design

The first steps are the preprocessing and the feature extraction during which the toolkit processes and adds information to the initial corpora available. Then, the most important step consists of training a classifier using the features extracted (training phase) or in the labelling of the test corpus (decoding phase).

We also added a threshold optimization and a feature selection phase which are later described (see Sections 5.5 and 6 respectively for threshold optimization and feature selection).

All these phases can be parameterized using a single configuration file.

4.2.1. Configuration file

A configuration file gathers the main WCE parameters. It is stored in YAML[5] format. The main configuration parameters concern the source and target languages involved and the path to the input corpus and its translation.

4.2.2. Preprocessing Phase

Preprocessing consists of obtaining POS tags, word alignments and all needed analyses from the available parallel corpus (the target being a MT output made up of raw text – 1-best and N-best of MT). First, input data is lowercased and/or tokenized if necessary. Then, TreeTagger toolkit [19] is applied to get the Part-Of-Speech (POS) tags and stem of each word in both source and target languages. The different POS extracted are normalized. Finally, word alignments are obtained using GIZA++ [20].

4.2.3. Features Extraction

As said before, the internal features come from the output of the Statistical Machine Translation (SMT) system. In this part we mainly focus on the extraction of the external features, given by toolkits which are not part of the SMT system.

The TreeTagger toolkit [19] is involved in the extraction of the following features: "Proper Names", "Unknown Stems" and "Source/Target Stem". GIZA++ [20] helps us to extract the context alignment features for POS, Word and Stems. To compute the features "Longest Target N-gram Length" and "Longest Source N-gram Length" we use the SRILM toolkit [21]. The word's constituent label ("Constituent Label") and its depth in the constituent tree ("Distance to Root") are also extracted using Bonsai (for French) [22, 23] or Berkeley parser (for other languages) [24]. To represent hierarchical structures and extract the two features, the Natural Language ToolKit (NLTK) [25] in Python is used. The BabelNet [26] API is used to extract the feature "Target polysemy count".

Finally, the features "Occurences in Google Translate" and "Occurences in Bing Translator" are extracted by using the *Google Translate* and *Bing Translator* API, respectively.

4.2.4. Training / Decoding Phase

Once the final feature extraction stage has been completed, we use Conditionnal Random Fields (CRF) as machine learning technique through the Wapiti toolkit [27].

The classifier uses all the chosen features and it is trained on a preliminary labelled corpus (see next section for example of corpora directly usable with our toolkit). During decoding phase, the classifier determines, from a test corpus, whether a word should be labelled as "correct" or "incorrect" (respectively *Good* or *Bad*).

[4] https://github.com/besacier/WCE-LIG
[5] http://www.yaml.org/

5. WCE Experiments

This section presents the experiments done for 2 different language pairs: French–English (*fr–en*) with the corpus provided by [28] and English–Spanish (*en–sp*) corresponding to the WMT shared task on word confidence estimation (2014 edition[6]).

5.1. The French–English post-edited corpus

The *fr–en* corpus contains 10881 translations. It was taken from several French–English news corpora from former WMT evaluation campaigns (from 2006 to 2010) [28].

To obtain the translations, [28] used a French–English phrase-based translation system based on the *Moses* toolkit [29]. This medium-sized system was trained on Europarl and News parallel corpora for a former WMT evaluation shared-task (system more precisely described in [30] - 1.6M parallel sentences and 48M monolingual sentences in target language).

The hypotheses translated were post-edited according to the methodology described in [28]. 10000 random sentences were extracted to create the training data and the remaining sentences were used for the evaluation corpus.

In order to evaluate our Word-level Confidence Estimation (WCE) system, we obtained a sequence q of quality labels (recall that $q = (q_1, q_2, ..., q_N)$ and $q_i \in \{good, bad\}$) using TER-Plus toolkit [31]. Each word or phrase in the hypothesis e_{hyp} is aligned to a word or phrase in the reference (e_{ref}) with different types of edit: "I" (insertions), "S" (substitutions), "T" (stem matches), "Y" (synonym matches), "P" (phrasal substitutions) and "E" (exact match). Then, 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.

An example of output of TER-Plus evaluation tool is shown in Table 3.

Original Ref.:	this	is	enough	to	shake	asset	prices	
Original Hyp.:	what	is	enough	to	cower	prices	of	assets
Ref.:	this	is	enough	to	*****	shake	asset	prices
Hyp.:	what	is	enough	to	cower	prices	of	assets
Hyp. After Shift:	what	is	enough	to	cower	of	assets	prices
Alignment:	S	E	E	E	I	S	T	E
Labels:	B	G	G	G	B	B	G	G

Table 3: Example of the TER-Plus toolkit's output processed

5.2. Adaptation to a new language pair

To evaluate our toolkit on another language pair (English–Spanish), we used the official data from WMT 2014 shared task on WCE.

One of the strength of our toolkit is the easiness to adapt it to another language pair within the (so-far) supported languages which are French, English, and Spanish. Thus, a few configuration parameters were changed to move from the French–English (*fr–en*) to English–Spanish (*en–es*), which are mainly the source language, the target language, and paths associated to input files.

Consequently, our WCE toolkit process *en–es* task in the same way as for *fr–en* task, but some features may not be extracted due to language-pair specificities: unavailable tools, no N-best, *etc.* For instance, for the *en–es* task, since the N-best list is not available, we cannot extract the five following internal features: "WPP Exact", "WPP Any", "Nodes", "WPP Min" and "WPP Max".

5.3. Results

The WCE evaluation measures are the Precision (P), the Recall (R) and the F-Measure (F) of each label (as reminder, the decision label can be either *good* or *bad*). We use *wapiti* [27] to train the CRF model and label the words.

5.4. Comparison with the State-of-the-Art

	Systems	***M-F***	***F***(*bad*)
	FBK-UPV-UEDIN-1 [32]	62.00	48.73
	LIMSI [33]	60.55	47.32
→	*Our toolkit*	*60.76*	*47.17*
	LIG-1 [9]	63.55	44.47
	LIG-2 [9]	63.77	44.11
	FBK-UPV-UEDIN-2 [32]	62.17	42.63

Table 4: Results of the best systems at the Word-level Quality Estimation task (*en–es*) at WMT14 [15], only the Mean F-Measure (*M-F*) and the F-Measure (*F*) on the bad labels are available to compare the performances of our toolkit.

Using the default decision threshold of our classifier, the Table 4 presents the results obtained in the WMT14 Quality Estimation shared task with the language pair English–Spanish (*en–es*).

The results show that our toolkit obtained similar performances compared to the State-of-the-Art. We could not compare with the CE toolkit mentionned in [12] since they did not provided full results within the framework of the WMT14 evaluation. Future work could involve a comparison between our toolkit and the toolkit presented in [12].

5.5. Decision threshold optimization

Table 5 shows the classification performances of our toolkit for the two different language pairs: the French–English (*fr–en*) and the English–Spanish (*en–es*). The latter corresponds to the Quality Estimation shared task of WMT14 [15].

Our toolkit proposes to optimize the decision threshold but, in this context, what we report can be only considered as an oracle threshold setting since no real development corpus was available for both language pairs. These results are only reported to demonstrate the ability of the toolkit to tune the decision threshold. With this optimization, the scores are improved for the *bad* label (+2.89 points) regarding the results obtained with the default threshold in the *fr–en* task. In the *en–es* task, the oracle threshold sightly improves the results, according to the Mean F-Measure (+0.11 points).

[6] http://www.statmt.org/wmt14/quality-estimation-task.html

Task	Threshold	Label	P	R	F	M-F
fr–en	*Default*	*Good*	84.45	90.22	**87.24**	64.96
		Bad	**50.10**	37.16	42.67	
	Optimized	*Good*	**85.60**	85.65	85.62	**65.59**
		Bad	45.61	**45.50**	**45.56**	
en–es (WMT14)	*Default*	*Good*	71.24	**77.73**	**74.35**	60.76
		Bad	**51.82**	43.28	47.17	
	Optimized	*Good*	**71.42**	76.82	74.03	**60.87**
		Bad	51.49	**44.45**	**47.71**	

Table 5: The toolkit's WCE performances with *fr–en* and *en–es* (WMT14) tasks. Note that for each language pair, the first block of results corresponds to the performance obtained with default decision threshold and the second block corresponds to the performance with an oracle threshold (to optimize Mean F-measure of *Good* and *Bad* labels).

6. Features selection

This section illustrates how the toolkit can be used for feature selection and analysis of performance with different feature sets. The next experiments reported were done for the *fr–en* task with the default decision threshold.

6.1. Experimenting with different feature sets

The following feature sets were evaluated in this section:

- the baseline features (*Base.*) given in Table 1 (not **"bold"**, not *"italic"*, no feat. 28-29),
- same as above + modified features estimated with a new method (in *"italic"* in Table 1) are added (*mod.*) ;
- same as above + the new features (*new*) mentionned in Table 1 (the ones in **"bold"**) ;
- same as above + features 28-29 of Table 1 involving online MT systems (*MT*).

Features	Labels	P	R	F	M-F
Base.	*Good*	81.97	**92.22**	86.80	58.64
	Bad	44.17	23.28	30.48	
+ *mod.*	*Good*	83.21	90.99	86.92	62.00
	Bad	47.24	30.53	37.09	
+ *new*	*Good*	83.55	90.11	86.70	62.65
	Bad	46.75	32.86	38.60	
+ *MT*	*Good*	**84.45**	90.22	**87.24**	**64.96**
	Bad	**50.10**	**37.16**	**42.67**	

Table 6: Improvements obtained regarding the features added. For both labels (Good and Bad) we use the Precision (P), Recall (R) and F-Measure (F). The Mean F-Measure of *Good* and *Bad* labels is presented in the last column.

We can observe for all the steps a general improvement of the Mean F-Measure in Table 6. The baseline is 58.64, while the use of modified features enables to get over 62. The new features show their usefulness with a Mean F-Measure score at 62.65 points. Finally, adding occurences coming from on-line Machine Translation systems enables us to get 64.96 points. Even if using online MT systems for WCE can appear as controversial, this seems to bring useful information to our classifier.

6.2. Feature selection using Sequential Forward Selection (SFS) algorithm

Going further, we propose to process a finer feature selection using the Sequential Forward Selection (SFS) algorithm for which scripts are made available in our toolkit distribution.

While feature selection can be made through several approaches [34], we chose to use the SFS method. It is a bottom up algorithm which starts from a feature set noted Y_k (which can be empty or not) and selects as first feature (x) the one that maximizes the Mean F-Measure, $MF(Y_k + x)$, from a set of features (J_k). The algorithm below summarizes the whole process:

while size of $J_k > 0$ **do**
  $maxval = 0$
  for $x \in J_k$ **do**
    if $maxval < MF(Y_k + x)$ **then**
      $maxval \leftarrow MF(Y_k + x)$
      $bestfeat \leftarrow x$
    end if
  end for
  add $bestfeat$ to Y_k
  remove $bestfeat$ from J_k
end while

In Table 7 we present the result of the SFS algorithm, which ranks our new features starting from an empty feature set. The dash line marks the limit of the best feature set according to the Mean F-Measure (with 65.14 points).

It appears that most of new features we added (in **"bold"**) bring relevant information associated to classical ones (no highlight and in *"italic"*). Only the feature "Target Stem" seems to be irrelevant for the *fr–en* task. One reason for that might be that for the English language, stem and words features may be highly correlated.

Rank	Feature	Rank	Feature
1	**Stem context Alignements**	16	Stop Words
2	WPP Exact	17	Nodes
3	*Word context Alignements*	18	**# of Stem Occurrences**
4	WPP Max	19	Numeric
5	WPP Any	20	**Unknown Stem**
6	WPP Min	21	Target Word
7	*POS context Alignements*	22	Source POS
8	Occur in Google Translate	23	*Polysemy Count – Target*
9	Longest Target N-gram Length	24	*Source Word*
10	**Occur in Bing Translate**	25	*Constituent Label*
11	**Source Stem**	26	Punctuation
12	Target Backoff Behaviour	27	**Target Stem**
13	Longest Source N-gram Length	28	*Proper Name*
14	# of Word Occurrences	29	Target POS
15	*Distance To Root*		

Table 7: Rank of each feature according to the Sequential Forward Selection algorithm within the framework of the *fr–en* task. The Dash line marks the best Mean F-Measure score obtained with 65.14 points.

This feature selection functionnality is provided with the toolkit, which means that whatever set of features the user wants to test, he/she can apply the SFS algorithm very easily.

7. Conclusion and Perspectives

This paper presented our Word Confidence Estimation (WCE) approach made available through an open-source toolkit. It combines some classical features as well as some new in-house features. All these features are passed through a Conditional Random Fields (CRF) classifier to estimate the correctness of a word.

The WCE experiments conducted achieve State-of-the-Art and reproducible performances measured on two different data sets corresponding to two language pairs (French–English and English–Spanish). Thanks to its flexibility, our toolkit is nearly language independent, as long as the user can provide grammars and models for the specified languages.

Our WCE toolkit has been packaged and released for others to be able to reproduce rapidly the experiments reported in this article. This package is made available on a *GitHub* repository[7] under the licence GPL V3.

In addition to this toolkit, comes a special module, which enables feature selection automatically using SFS algorithm (sequential forward selection). A more performant algorithm will be added in the near future like the Sequential Floating Forward Selection algorithm, which has backtracking capabilities.

Further work will focus on (i) adding features (based on word embeddings for instance) and (ii) evaluating the toolkit efficiency in a real Computer Assisted Translation (CAT) framework. We also plan to extend our toolkit to the design of WCE for speech recognition and speech translation tasks.

8. Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

This work was partially founded by the French National Research Agency (ANR) through the KEHATH Project.

9. References

[1] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proceedings of 18th International Conference on Machine Learning*. Morgan Kaufmann, San Francisco, Californie, États-Unis d'Amrique, 2001, pp. 282–289.

[2] N.-Q. Luong, L. Besacier, and B. Lecouteux, "Towards Accurate Predictors of Word Quality for Machine Translation: Lessons Learned on French - English and English - Spanish Systems," *Data and Knowledge Engineering*, p. 11, Apr. 2015.

[3] 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.

[4] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing, "Confidence estimation for machine translation," in *Proceedings of COLING 2004*, Geneva, April 2004, pp. 315–321.

[5] F. Wessel, R. Schlüter, K. Macherey, and H. Ney, "Confidence Measures for Large Vocabulary Continuous Speech Recognition," *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING*, 2001.

[6] D. Xiong, M. Zhang, and H. Li, "Error Detection for Statistical Machine Translation Using Linguistic Features," in *Proceedings of the 48th Association for Computational Linguistics*, Uppsala, Sweden, July 2010, pp. 604–611.

[7] M. Felice and L. Specia, "Linguistic Features for Quality Estimation," in *Proceedings of the 7th Workshop on Statistical Machine Translation*, Montreal, Canada, June 7-8 2012, pp. 96–103.

[8] 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, Aug. 2013, pp. 365–372.

[9] N.-Q. Luong, L. Besacier, and B. Lecouteux, "LIG System for Word Level QE task at WMT14," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland USA, June 2014, pp. 335–341.

[10] D. Langlois, S. Raybaud, and K. Smaïli, "Loria system for the WMT12 quality estimation shared task," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*, Baltimore, Maryland USA, June 2012, pp. 114–119.

[11] 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.

[12] L. Specia, G. Paetzold, and C. Scarton, "Multi-level Translation Quality Prediction with QuEst++," in *53rd Annual Meeting of the Association for Computational Linguistics and Seventh International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing: System Demonstrations*, 2015, pp. 115–120.

[13] C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2012 Workshop on Statistical Machine Translation," in *Proceedings of the Seventh Workshop on Statistical Machine*

[7] https://github.com/besacier/WCE-LIG

Translation. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 10–51.

[14] O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2013 Workshop on Statistical Machine Translation," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 1–44.

[15] 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, Maryland, USA: Association for Computational Linguistics, June 2014, pp. 12–58.

[16] S. Raybaud, D. Langlois, and K. Smaili, ""This sentence is wrong." Detecting errors in machine-translated sentences," *Machine Translation*, vol. 25, no. 1, pp. p. 1–34, Aug. 2011.

[17] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *NIPS*, 2013.

[18] 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 2011, pp. 211–219.

[19] H. Schmid, "Improvements in Part-of-Speech Tagging with an Application to German," in *Proceedings of the ACL SIGDAT-Workshop*, Dublin, Ireland, 1995.

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

[21] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," in *Seventh International Conference on Spoken Language Processing*, Denver, USA, 2002, pp. 901–904.

[22] A. Laurent, N. Camelin, and C. Raymond, "Boosting bonsai trees for efficient features combination: application to speaker role identification," in *InterSpeech*, Singapour, September 2014, pp. 76–80.

[23] M. Candito, J. Nivre, P. Denis, and E. H. Anguiano, "Benchmarking of Statistical Dependency Parsers for French," in *Proceedings of COLING'2010*, 2010.

[24] S. Petrov and D. Klein, "Improved Inference for Unlexicalized Parsing," in *HLT-NAACL*, 2007.

[25] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.

[26] R. Navigli and S. P. Ponzetto, "BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network," *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.

[27] T. Lavergne, O. Cappé, and F. Yvon, "Practical Very Large Scale CRFs," in *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, July 2010, pp. 504–513.

[28] M. Potet, E. Esperança-Rodier, 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.

[29] 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, June 2007, pp. 177–180.

[30] 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.

[31] M. Snover, N. Madnani, B. Dorr, and R. Schwartz, "TERp system description," in *MetricsMATR workshop at AMTA*, 2008.

[32] J. G. Camargo de Souza, J. González-Rubio, C. Buck, M. Turchi, and M. Negri, "FBK-UPV-UEdin participation in the WMT14 Quality Estimation shared-task," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, June 2014, pp. 322–328.

[33] G. Wisniewski, N. Pécheux, A. Allauzen, and F. Yvon, "LIMSI Submission for WMT'14 QE Task," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, June 2014, pp. 348–354.

[34] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern recognition*, vol. 33, no. 1, pp. 25–41, 2000.

Improving Translation of Emphasis with Pause Prediction in Speech-to-speech Translation Systems

Quoc Truong Do[⋆]*,Sakriani Sakti*[⋆]*, Graham Neubig*[⋆]*, Tomoki Toda*[†]*, Satoshi Nakamura*[⋆]

[⋆] Nara Institute of Science and Technology, Japan
{do.truong.dj3,neubig,ssakti,s-nakamura}@is.naist.jp
[†] Nagoya University, Japan
tomoki@icts.nagoya-u.ac.jp

Abstract

Prosodic emphasis is a vital element of speech-based communicating, and machine translation of emphasis has been an active research target. For example, there is some previous work on translation of word-level emphasis through the cross-lingual transfer of F_0, power, or duration. However, no previous work has covered a type of information that might have a large potential benefit in emphasizing speech, pauses between words. In this paper, we first investigate the importance of pauses in emphasizing speech by analyzing the number of pauses inserted surrounding emphasized words. Then, we develop a pause prediction model that can be integrated into an existing emphasis translation system. Experiments showed that the proposed emphasis translation system integrating the pause prediction model made it easier for human listeners to identify emphasis in the target language, with an overall gain of 2% in human subjects' emphasis prediction F-measure.

1. Introduction

Emphasis is an important factor of human communication that conveys the focus of speech. For example, in our daily life, it is common for words to be misheard in many situations, particularly in noisy environments. When such a situation happens, people often put more emphasis (focus) on particular words that are misheard to help listeners understand which information in the sentence is the most important. Emphasis is as important, or even more important in cross-lingual communication because of the need for understanding the main ideas of people speaking in different languages despite the barriers posed by cross-lingual communication.

Speech-to-speech (S2S) translation [1] is a technique that is able to translate speech across languages as illustrated in Fig. 1. In order to convey emphasis across languages, several previous works [2, 3] have proposed methods to translate emphasis in a limited domain, 10 digits. Anumanchipalli et al. [4] translates emphasis in a larger domain, but only consider F_0 features. Do et al. [5] take a different approach of translating emphasis by considering emphasis as a real-numbered

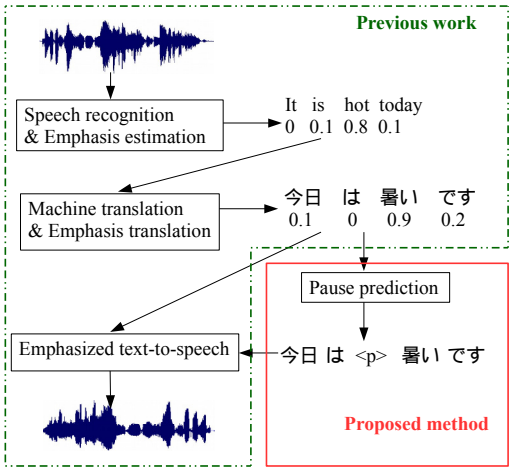


Figure 1: Proposed method for predicting pauses and using them in the translation of emphasis. Pauses are represented in text as "<p>".

value and utilizing all speech features including F_0, duration, and power. However, all these methods are still missing a variety of information that might have a large potential benefit in emphasizing speech: pauses.

Pauses are one of the prosodic cues that segment speech into meaningful units [6]. In emphasized speech, along with power, duration, and F_0, we conjecture that pauses also are used to indicate that upcoming words are important and give a sign to listeners that they should pay attention to those words. However, the previous works on emphasis modeling and emphasis translation have not analyzed the importance of pauses in emphasized speech, and not incorporated them into the translation of emphasis in S2S translation systems.

In this paper, we first perform an analysis to investigate the importance of pauses in emphasizing speech by looking at the number of pauses inserted surrounding emphasized words in English and Japanese, and examine the relationship of pause usage between those two languages. Then, based on this knowledge, we investigate the contribution of incorporating an automatic pause prediction system into an existing method for translating emphasis in S2S translation, as illus-

trated in Fig. 1.

2. Emphasis in speech-to-speech translation

This section describes a S2S translation framework that is able to convey emphasis across languages [5]. The "previous work" section in Fig. 1 (inside the green box) is broken down in more detail in Fig. 2.

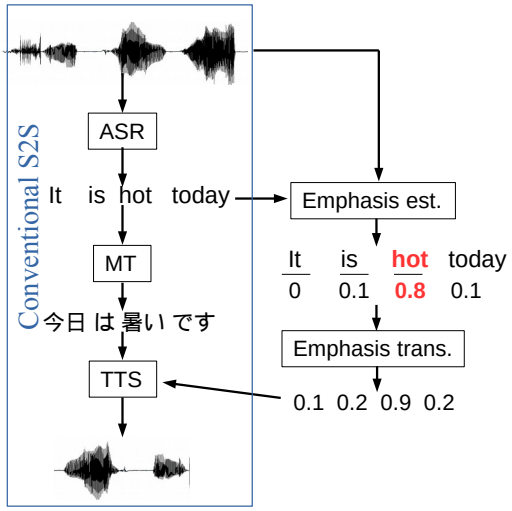


Figure 2: A S2S translation system capable of translating emphasis, consisting of a conventional S2S system, emphasis estimation, and an emphasis translation system.

2.1. Conventional speech-to-speech translation systems

Conventional S2S translation systems have been studied extensively in previous works, such as [1, 7]. As illustrated in Fig. 2, they consist of 3 main components: speech recognition recognizes speech into text, machine translation translates the text into the target language, and text-to-speech synthesizes speech given the translated text. Recently, many approaches have been proposed to improve the performance of S2S systems, for instance, [8] proposed an interesting idea that detects errors in ASR and MT output, then asks users to clarify the speech before translation.

Although the performance of conventional S2S systems is improving in conveying the meaning of speech, they are still lack of paralinguistic information, particularly emphasis.

2.2. Emphasis estimation

In order to translate emphasis, the first step is to extract information that representing emphasis. [5] has applied linear-regression hidden semi-Markov models, which are a simple form of multi-regression HSMMs [9] to derive a real-numbered value called word-level emphasis degree that represents how emphasized a word is. Defining the approach mathematically, given a word sequence consisting of N words and its speech features $\boldsymbol{o}$, a sequence of N word-level emphasis values $\boldsymbol{\Lambda} = [\lambda_1, \cdots, \lambda_N]$ is derived by maximizing a likelihood function

$$P(\boldsymbol{o}|\boldsymbol{\lambda}, \mathcal{M}) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{q}|\boldsymbol{\lambda}, \mathcal{M})\, P(\boldsymbol{o}|\boldsymbol{q}, \boldsymbol{\lambda}, \mathcal{M}), \qquad (1)$$

where $\boldsymbol{q}$ is a HMM state sequence that corresponds to the given word sequence, and $\mathcal{M}$ is the model parameters. This approach has the advantage that all features that are used to emphasize words such as power, F_0, and duration are taken into account, while other works on emphasis translation only utilized individual features separately [4, 10].

2.3. Emphasis translation

As described in [5], the word-level emphasis sequence is translated across languages by utilizing conditional random fields (CRFs) [11]. The problem is defined as follows: given a source language word sequence $\boldsymbol{w}^{(f)}$, a vector of word-level emphasis $\boldsymbol{\Lambda}^{(f)}$, a corresponding target word sequence $\boldsymbol{w}^{(e)}$ (which is the output of the MT system), and part-of-speech tag information $\{\boldsymbol{t}^{(e)}, \boldsymbol{t}^{(f)}\}$, we want to predict the target language word-level emphasis vector, as illustrated in Fig. 3. The probability of the target word-level emphasis sequence $\boldsymbol{\Lambda}^{(e)}$ is calculated by

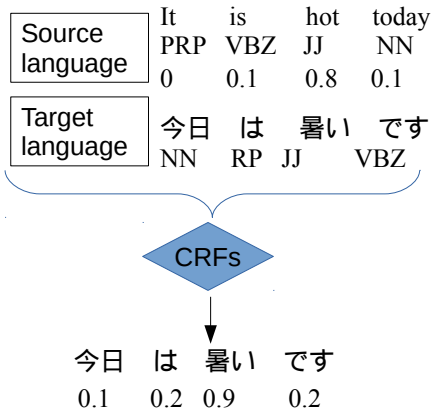


Figure 3: CRF-based emphasis translation.

$$P(\boldsymbol{\Lambda}^{(e)}|\mathbf{x}) = \frac{\prod_{n=1}^{N} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(\lambda_{n-1}^{(e)}, \lambda_n^{(e)}, \mathbf{x}_n^{(k)})\right\}}{\sum_{\tilde{\boldsymbol{\lambda}}^{(e)}} \prod_{n=1}^{N} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(\tilde{\lambda}_{n-1}^{(e)}, \tilde{\lambda}_n^{(e)}, \mathbf{x}_n^{(k)})\right\}}, \qquad (2)$$

where $\boldsymbol{x}$ is the input features, f is feature functions, K is the number of feature functions, and θ is the model parameters. The advantage of CRF-based translation model is that it flexible, and easy to add more features or remove irrelevant features that are not helpful for translation.

3. Pause prediction

Pause prediction is not a new research field, with a large body of research trying to tackle this problem [12, 13, 14]. The main distinction between these previous methods and our work is that while previous methods attempted to predict pauses from text (linguistic) information only, in our work we are given information about whether the word in question is emphasized, which gives us a stronger signal about whether pauses should be inserted or not. In this section, we describe two approaches that are able to utilize both linguistic and emphasis information to predict pauses based on CRFs.

The pause prediction problem can be described as follows: Given a word sequence and its word-level emphasis sequence, we want to predict in which of the below 4 positions a pause is inserted.

Before : a pause is inserted before the word.

After : a pause is inserted after the word.

Both sides : pauses are inserted before and after the word.

None : there is no pause inserted.

Generally speaking, this is a classification problem with 4 classes.

3.1. Pause extraction

The first step is to extract pauses from the training data by 3 steps, first, we train a speech recognition model on the same data, this step will give us a speaker dependent acoustic model for each speaker. Then, we perform forced alignment on the training data to derive audio-text alignments. Finally, from the alignment, we extract all pause segments that have duration at least 50ms as pauses.

3.2. CRF-based pause prediction

The CRF-based prediction model is very similar to emphasis translation described in Section 2.3. The input features include words, part-of-speech tags, emphasis degree, and context information of the preceding and succeeding units. Table 1 shows an example of input features. In the example, the word *hot* is the emphasized word, and we can see that a pause is inserted after the word *is* and before the word *hot*. In a standard sentence, this placement of a pause may seem unnatural. However, because the word *hot* is emphasized intentionally, the pause can be inserted to give a sign that the word *hot* is important.

4. Experiments

4.1. Experimental setup

The experiments were conducted using a bilingual English-Japanese emphasized speech corpus [15], which has emphasized content words that were carefully selected to maintain the naturalness of emphasized utterances. The corpus consists of 966 pairs of utterances with 1258 emphasized and 3886 normal words. The speech data is collected from 3 bilingual speakers, 6 monolingual Japanese, and 1 monolingual English speaker. The training data is divided into 916 training and 50 testing samples. And the setup for emphasis translation follows our previous work [5], extracting speech features using 25-dimension mel-cepstral coefficients including spectral parameters, log-scaled F_0, and aperiodic features. Each speech parameter vector includes static features and their delta and delta-deltas. The frame shift was set to 5 ms. Each HSMM model is modeled by 7 HMM states including initial and final states. We adopt STRAIGHT [14] for speech analysis.

Table 1: An example of input features for the sentence "it is <p> hot" with word-level emphasis sequence "0 0.1 0.8". Note that pauses are represented by commas, and we also use the context information of the preceding and succeeding units.

Position	Word	Part-of-speech	Emphasis
None	it	PRP	0
After	is	VBZ	0.1
Before	hot	JJ	0.8

4.2. Pause insertion analysis

In the first experiment, we investigate the importance of pause insertion in emphasizing words by analyzing number of pauses inserted before, after, and on both sides of emphasized words. The result is shown in Table 2.

First, we look at the column data indicating the number of pauses insertions in each position. We can easily see that the number of pauses inserted after emphasized words is dominant among all subjects and languages, and it is not common that pauses are inserted on both sides of emphasized words. This indicates that in order to emphasize words, the speaker often insert a pause after the emphasized word, and this usage is independent of whether the language is English or Japanese.

Second, comparing the number of pause insertions between English and Japanese at lines 1-2, 3-6, and 4-5, we can see that the difference is small in the "Before" position; but much a larger in the "after" and "both sides" positions, in which Japanese has more pause insertion than English.

Moreover, an analysis on pause insertions surrounding normal words for native speakers is also conducted as showed in Table 3. We can see that there is a small number of pauses inserted surrounding normal words, this is likely normal words are less likely to induce pauses, and also because the utterances are relatively short, ranging from 4 to 16 words.

According to above observations, we conclude that 1) pauses are an important factor in both languages that

helps to express emphasis, and 2) it is better to consider pause insertion in an emphasis translation system between English-Japanese, especially when translating from English to Japanese because pauses are even more often used in Japanese than English.

Table 2: Number of pauses inserted corresponding to different positions surrounding emphasized words. "All [English|Japanese]" denotes the case where we use all data including native and non-native speakers.

	Before	After	Both sides
1. All English	117	230	33
2. All Japanese	125	499	241
3. English by natives	155	**248**	**48**
4. English by non-natives	42	194	3
5. Japanese by non-natives	178	337	113
6. Japanese by natives	104	**564**	**292**

Table 3: Number of pauses inserted corresponding to different positions surrounding normal words.

	Before	After	Both sides
1. English by natives	47	44	1
2. Japanese by natives	167	182	6

4.3. Pause insertion prediction

In the next experiment, we evaluate the performance of pause prediction models based on CRFs. 4 classes were used, they are "none", "before", "after", and "both sides". The corpus is divided into 2 sets of 916 training and 50 testing utterances from one native Japanese speaker. We used a single speaker because the pause prediction system will be integrated into an existing emphasis S2S translation system that is speaker-dependent.

We evaluate the performance of the CRF-based pause prediction model using different combination of input features, which includes words, part-of-speech tags, word-level emphasis degree, and information of preceding and succeeding units. The measurement metric is F-measure, which is the harmonic mean of precision and recall. The result is shown in Table 4.

Table 4: Pause prediction performance using different combination of input features. "ctx" denotes context information of a preceding and succeeding units.

Emph.	Emph. ctx.	Word	Word ctx.	Tag	Tag ctx.	F-measure
✓	✓	✓	✓	✓	✓	**88.76**
		✓	✓	✓	✓	85.38
				✓	✓	84.81
✓		✓		✓		85.71

First, by comparing the 1st line with the 2nd and 3rd line. We can see that emphasis information is important for pause prediction, improving 3% F-measure. Second, the last line that shows the input feature without context information has lower accuracy compared to the 1st line, which has context information, indicating that the context information is also very important because it gives more information for pause prediction.

4.4. Emphasis translation with pause insertion

In the final experiment, we evaluate the S2S translation system integrating with the CRF-based pause prediction model. Four systems were:

No-emphasis : A speech translation system without emphasis translation as described in [5].

Baseline : An emphasis translation system without pause prediction as described in [5].

+Pause : The baseline system with the CRF-based pause prediction model.

Natural : Natural speech by native Japanese speaker.

First, we synthesize audios from each system. Then, we asked 6 native Japanese listeners to listen to the synthesized audio and identify the emphasized word. Finally, we score each system with F-measure. In addition, we perform an objective evaluation where the emphasized word is detected by an emphasis threshold of 0.5[1] yielding 91.6% F-measure. Note that it is not possible that the subjective result is better than the objective result, because there is a chance that text-to-speech systems make mistakes in synthesizing emphasized audios. The result is shown in Fig. 4.

As reported in [5], the baseline system outperforms *No-emphasis* system in conveying emphasis across languages. However, it is still 4% lower accuracy than the objective evaluation. By integrating the pause prediction model, we gain 2% F-measure, which is closer to the objective result. The result indicates that pauses are an important type of information that helps listeners perceive the focus of speech better, and also prove our conjecture that pause might be used to indicate that upcoming words are important.

5. Conclusion

In this paper, we investigated the importance of pauses in emphasizing speech, as well as integrating a pause prediction model – that utilized both linguistic and emphasis features – into an existing emphasis translation system. Results of an analysis and emphasis translation experiments from English to Japanese show that 1) pauses are important type of information in that helps listeners better perceive the focus of speech, 2) along with linguistic features, we found that emphasis features also plays an important role in predicting

[1] This value is an optimized value that has been tested in [5].

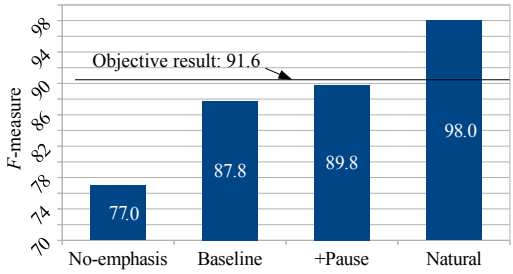


Figure 4: Subjective evaluation of emphasis translation with pause insertion.

pauses in emphasized speech, and 3) the emphasis translation system achieves a 2% F-measure improvement with a pause prediction model. Future works will examine more pause prediction models, and also analyze pause usage in more languages.

6. Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Number 24240032 and by the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan.

7. References

[1] S. Nakamura, "Overcoming the language barrier with speech translation technology," *Science & Technology Trends - Quarterly Review No.31*, April 2009.

[2] T. Kano, S. Takamichi, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "Generalizing continuous-space translation of paralinguistic information," in *Proceedings of Interspeech*, August 2013.

[3] T. Kano, S. Sakti, S. Takamichi, G. Neubig, T. Toda, and S. Nakamura, "A method for translation of paralinguistic information," in *Proceedings of IWSLT*, December 2012, pp. 158–163.

[4] G. Anumanchipalli, L. Oliveira, and A. Black, "Intent transfer in speech-to-speech machine translation," in *Proceedings of SLT*, Dec 2012, pp. 153–158.

[5] Q. T. Do, S. Takamichi, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "Preserving word-level emphasis in speech-to-speech translation using linear regression HSMMs," in *Proceedings of InterSpeech*, September 2015.

[6] S. Dowhower, "Speaking of prosody: Fluency's unattended bedfellow," *Theory into Practice*, vol. 30, no. 3, pp. pp. 165–175, 1991.

[7] G. Kumar, M. Post, D. Povey, and S. Khudanpur, "Some insights from translating conversational telephone speech," in *Proceedings of ICASSP*, May 2014, pp. 3231–3235.

[8] N. Ayan, A. Mandal, M. Frandsen, J. Zheng, P. Blasco, A. Kathol, F. Bechet, B. Favre, A. Marin, T. Kwiatkowski, M. Ostendorf, L. Zettlemoyer, P. Salletmayr, J. Hirschberg, and S. Stoyanchev, "Can you give me another word for hyperbaric?: Improving speech translation using targeted clarification questions," in *Proceedings of ICASSP*, May 2013, pp. 8391–8395.

[9] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi, "A style control technique for HMM-based expressive speech synthesis," *Transactions on IEICE*, vol. E90-D, no. 9, pp. 1406–1413, Sept. 2007.

[10] P. Aguero, J. Adell, and A. Bonafonte, "Prosody generation for speech-to-speech translation," in *Proceedings of ICASSP*, vol. 1, 2006.

[11] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of ICML*, 2001, pp. 282–289.

[12] T. T. Nguyen, G. Neubig, H. Shindo, S. Sakti, T. Toda, and S. Nakamura, "A latent variable model for joint pause prediction and dependency parsing," in *Proceedings of InterSpeech*, Dresden, Germany, September 2015.

[13] V. Sridhar, J. Chen, S. Bangalore, and A. Conkie, "Role of pausing in text-to-speech synthesis for simultaneous interpretation," in *Proceedings of ISCA Workshop on Speech Synthesis*, 2013.

[14] J. Tauberer, "Predicting intrasentential pauses: Is syntactic structure useful?" in *Proceedings of the Speech Prosody*, 2008, pp. 405–408.

[15] D. Q. Truong, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "Collection and analysis of a Japanese-English emphasized speech corpus," in *Proceedings of Oriental COCOSDA*, September 2014.

Evaluation and Revision of a Speech Translation System for Healthcare

Mark Seligman and Mike Dillinger

Spoken Translation, Inc.
mark.seligman@spokentranslation.com
mike.dillinger@gmail.com

Abstract

Earlier papers have reported on Converser for Healthcare, a highly-interactive English↔Spanish speech translation system for communication between patients and caregivers, and upon an extensive pilot project testing the system at a San Francisco medical center, part of a very large healthcare organization. This historical paper provides for the first time details of the resulting evaluation and fully describes the associated system revisions to date.

1. Introduction

Spoken language translation systems are now in operation at Google and Microsoft/Skype, and multiple applications for spoken language translation (SLT) or automatic interpreting are also available – SpeechTrans, Jibbigo, iTranslate, and others. However, widespread use remains in the future for serious use cases like healthcare, business, emergency relief, and law enforcement, despite demonstrably high demand.

In spite of dramatic advances during the last decade, both speech recognition and translation technologies are still error-prone. While the error rates may be tolerable when the technologies are used separately, the errors combine and even compound when they are used together. The resulting translation output is often below the threshold of usability when accuracy is essential. As a result, present use is still largely restricted to use cases – social networking, travel – in which no representation concerning accuracy is demanded or given.

The speech translation system discussed here, Converser for Healthcare, applies interactive verification and correction techniques to this essential problem of overall reliability.

Earlier papers ([1], [2], [4], [5], [6], [7], [8], [9]) have reported on this highly-interactive system for English↔Spanish communication between patients and caregivers, and upon an extensive pilot project in 2011 testing Version 3.0 of the system at a San Francisco medical center, part of a very large healthcare organization ([9]). This paper provides for the first time details of the resulting evaluation and fully describes the associated system revisions to date, yielding the current Version 4.0. The paper is partly of historical interest, since the pilot took place four years ago – a long time in computer years. However, most of the issues raised by the evaluation remain current, and will be discussed below.

For orientation, Section 2 of this paper will review Converser's basic interactive facilities, as common to both Versions 3.0 and 4.0. Section 3 gives the results of the pilot project, as seen in the independent evaluation commissioned by the healthcare organization. Section 4 then details the revisions which were made for Version 4.0 in response to this feedback and other lessons learned. Section 5 offers an extended example of the revised system in use. We conclude in a final section.

2. The Converser System

We now briefly describe Converser's approach to interactive automatic interpretation, restricting description to core elements common to Version 3.0 (as used in the pilot project discussed in Section 3) and to the revised Version 4.0 (to be described in Sections 4 and 5 below). We'll concentrate on the system's verification/correction and customization features.

First, users can monitor and correct the speech recognition system to ensure that the text which will be passed to the machine translation component is completely correct. Speech, typing, or handwriting can be used to repair speech recognition errors.

Next, during the machine translation (MT) stage, users can monitor, and if necessary correct, one especially important aspect of the translation – lexical disambiguation.

The system's approach to lexical disambiguation is twofold: first, we supply a *back-translation*, or re-translation of the translation. Using this paraphrase of the initial input, even a monolingual user can make an initial judgment concerning the quality of the preliminary machine translation output. Other systems, e.g. IBM's MASTOR ([4]), have also employed re-translation. Converser, however, exploits proprietary technologies to ensure that the lexical senses used during back-translation accurately reflect those used in forward translation.

In addition, if uncertainty remains about the correctness of a given word sense, the system supplies a proprietary set of Meaning Cues™ – synonyms, definitions, etc. – which have been drawn from various resources, collated in a database (called SELECT™), and aligned with the respective lexica of the relevant MT systems. With these cues as guides, the user can monitor the current, proposed meaning and when necessary select a different, preferred meaning from among those available. Automatic updates of translation and back-translation then follow.

The initial purpose of these techniques is to increase reliability during real-time speech translation sessions. Equally important, however, they can also enable even monolingual users to supply feedback for off-line machine learning to improve the system. Until now, only users with some knowledge of the output language have been able to supply such feedback, e.g. in Google Translate.

Converser adopts rather than creates its speech and translation components, adding value through the interactive interface elements to be explained. Nuance, Inc. supplies speech recognition; rule-based English↔Spanish machine translation is supplied by Word Magic of Costa Rica; and text-to-speech is again provided by Nuance.

The Converser system includes Translation Shortcuts™ – pre-packaged translations, providing a kind of translation memory. When they're used, re-verification of a given utterance is unnecessary, since Shortcuts are pre-translated by professionals (or, in future versions of the system, verified using the system's feedback and correction tools). Access to stored Shortcuts is very quick, with little or no need for text entry. *Shortcut Search* can retrieve a set of relevant phrases given only keywords or the first few characters or words of a string. (If no Shortcut is found to match the input text, the system seamlessly gives access to broad-coverage, interactive speech translation.) A **Translation Shortcuts Browser** is provided (on the left in Figure 1), so that users can find needed phrases by traversing a tree of Shortcut categories, and then execute them by tapping or clicking. Shortcuts are fully discussed in [8].

Identical facilities are available for Spanish as for English speakers: when the Spanish flag is clicked, all interface elements – buttons and menus, onscreen messages, Translation Shortcuts, handwriting recognition, etc. – change to Spanish.

3. Pilot Project and Evaluation

We now turn to a pilot project which tested Converser for Healthcare, Version 3.0, in three departments (Pharmacy, Inpatient Nursing, and Eye Care) of a large hospital complex belonging to a major US healthcare organization.

The hardware and software used in the project have been described and assessed in [6]. Accordingly, our focus here will be on the user experience. We rely on the healthcare organization's internal report, based on a commissioned survey by an independent third party, an experienced medical interpreter from an accredited local agency. While the report itself is proprietary, we'll reproduce its findings in essence.

First, however, several preliminary points are in order concerning stumbling blocks for the pilot project. As we will see below, all of these impediments have by now been removed as a result of the striking infrastructure advances over the four years since the pilot concluded.

Converser Version 3.0 was designed to cooperate with the then-current Dragon NaturallySpeaking, to be installed separately, and thus required *speaker-dependent* speech recognition: each speaker had to register his or her voice. This process took two or three minutes, including a 30-second speech sample; and, while this interruption was no great burden for English-speaking staff members, it usually made speech recognition from the Spanish patients' side impractical.

Microsoft's handwriting recognition was integrated into the system for both languages; but correction of errors was tricky at the time, so that this addition, too, incurred a training cost.

One more speed bump resulted from a software feature intended for customization: patients and staff could be registered in Converser, so that their names could appear in transcripts, and so that various personalization features could be added later. However, registration of the login user was required rather than optional; and this process necessitated still more training time.

Taken together, these obstacles necessitated 45-minute training sessions for participating staff members.

Further, because the experiments predated the era of modern tablets, portability was inferior to that available now, while physical set up was much less convenient ([9]). On the first-generation tablets used, for instance, it was necessary to manually configure the physical buttons which turned the microphone on and off.

With these initial obstacles in mind, we can now review the results of the organization's evaluation.

Project goals. The organization's goals for the project were stated in terms of the problem to be solved, as follows: (Throughout, we closely paraphrase the original language of the report.)

- Members' [i.e. patients'] language needs remain unmet in many situations throughout the … organization. Since the needs vary from situation to situation, no single solution can be expected.
- Different interpretative solutions need to be tested and analyzed to determine their best fit on multiple variables such as setting, situation, type of patient, etc.
- Accuracy of translation and member acceptance of technology-based interpretive services vs. in-person interpretation need to be assessed.

The independent interviewer observed 61 real-time translation interactions – some involving spoken input, some with typed or handwritten input – and solicited reactions from most of the staff and patients involved. (A few patients declined to answer the questions.) Interviews included both open-ended requests for reactions and prepared questions.

Patients' reactions. Positive comments from patients included the following:

- "cool"
- "useful" – **5 mentions**
- "looks good" "well done"
- "would help"
- "good tool" – **2-3 mentions**
- "I would recommend it"
- Even if translation was not 100%, it was always understood
- "Perfect and clear" – **2 mentions**
- Saving time – don't have to wait for an interpreter
- "I like it"
- "I like the idea of it"
- Good for emergencies – **2 mentions**

Less positive or negative comments included these:

- GUI too complicated (need larger buttons, crowded screen, …) – **6 mentions**.
- Literacy issues: some immigrants can't read or write – **6 mentions**
- Font size too small – **3 mentions**
- "Too technical for me" "I don't like computers": family say elderly can't use – **8 mentions**
- Quality of Sound/Volume issues – **6 mentions**
- Handwriting didn't work – **6 mentions** (Note: usage was limited)
- Worries about quality of translation – **2 mentions**
- Keyboard issues (hard to use, pen is faster …) – **5 mentions**
- Problems with English voice – **2 mentions**
- System slow or froze – **6 mentions**
- Hard to use tablet in hospital – **1-2 mentions**

Some general patient comments:

- Training (for users) would be needed – **4 mentions**
- Product would be "ideal" with voice recognition – **4 mentions**
- A lot of mixed comments – They like the system but worry others (elderly, less literate) will struggle with it. (These comments came largely from partial or full English speaking members.)
- Would rather have an in person interpreter – **4-5 mentions**

Staff reactions. Positive staff comments:

- Good for short interactions
- Writing was easier than talking
- Typing was easier than talking
- You can verify translations better vs. Language Line – **2-3 mentions**
- I would use it if no other options
- Portability is good

Less positive or negative staff comments:

- Occasionally missed a sentence
- Computer literacy of members is a real issue. – **3 mentions** (Also, elderly can't double-click fast enough.)
- User Interface – buttons crowded
- Translations were a bit odd
- Slow
- Hard for patients to write on the tablet in bed – **2 mentions**
- Takes valuable time for the system to process

General staff comments:

- Training of patient's voice for DragonNaturallySpeaking would be needed.
- But time is limited already (i.e. no time in visit to train patients) – **4 mentions**
- Training for staff and providers needed – **3 mentions**
- This product is really more needed for Cantonese/Mandarin here in San Francisco.
- The system needs a formal introduction (so that the system can describe itself: for English providers to use with Spanish members).

Summary. Overview of patient and staff evaluations:

- High praise for the "idea." Higher than the actual experience of it
- Translation quality definitely "good enough" as rated by Members/Patients
- Limited English speakers (who can get along) would still use to verify the conversation and ensure completeness.
- Issues of literacy and computer literacy impact applicability
- Even though the system had issues (low to fair GUI, slow processing, lack of recognition of voice etc.), members partial or full English speakers thought it was "cool."
- Most people, and especially those who lacked English skills, preferred an in- person interpreter, although one person noted it saves time waiting for an interpreter, and a provider commented it saved the wait for Language Line.
- Good for emergencies
- Hard for members to use tablet in the hospital
- A number of patients declined to use in hospital but we lack data as to why.

Patient responses to six significant questions are tabulated in Table 1. The rightmost column shows the percentage of respondents who replied to each question with Completely or Mostly.

Most significantly, when asked whether the system met their needs, of the 79% of interviewed patients who answered the question, 94% responded either Completely or Mostly.

Table 1: **Patient responses to six questions**

Patient Evaluation	**% Answered Question**	**Completely or Mostly**
Did this meet your needs?	79%	94%
Was it accurate?	79%	90%
Was it easy to use?	72%	57%
Prefer handwriting question	67%	68%
Prefer using keyboard	67%	17%
Prefer to use handwriting and keyboard	67%	12%

4. System Revision

Having conveyed the organization's own assessment of the Converser for Healthcare 3.0 pilot project, we go on to describe the revisions prompted by it.

First and foremost, there was a glaring need to facilitate speech input from the Spanish side. This goal implied implementation of *speaker-independent* speech recognition; and this has been carried out by exploiting advances in Dragon NaturallySpeaking. Auxiliary third-party software has also been required to enable adaptation of Dragon software for use on desktop and tablet computers.

The need was also obvious for reduction in setup and training time. The following improvements reduce total warm-up to a few minutes for both staff and patients.

- The requirement for registration of the login user has been relaxed: registration is now optional, so that users can begin using the system immediately at startup time.
- An on-screen microphone button has now been substituted for the physical buttons previously used, so button configuration is no longer needed.
- Microsoft handwriting recognition has improved to the point that its correction facilities can be learned independently. Likewise, the company's on-screen keyboard now supports larger keys, so that on-screen typing has become more practical.
- Delivery of Converser via the Web will be enabled, so that only installation of the client software, providing access to a virtual desktop, will be required.

Another clear need has been to speed the interactions. While numerous staff members (and, separately, their managers) praised the ability to verify translations, others also stressed that verification consumed limited time. To balance these competing wishes, we have implemented a new set of icons allowing quick switching between Pre-Check and Post-Check modes. In the latter mode, useful when speed is more important than accuracy, speech recognition and translation are not checked in advance of transmission; but *post*-verification is still enabled, since back-translations are still generated and now appear in the bilingual transcripts (see Section 5). A **Rewind Button** has been supplied as well, so that erroneous or unsatisfactory translations can be quickly repaired and retransmitted. These new controls operate separately for English and Spanish speakers, so that, for instance, a doctor can pre-check when appropriate while allowing the patient to respond without distractions.

A number of interviewees called for various improvements in the user interface. In response, we have supplied large fonts for all on-screen elements (the exact size can be selected); added prominent icons for easier switching between English and Spanish speakers; enabled adjustment of the text-to-speech volume and speed, for easier comprehension; and added a quick way for staff to introduce Converser to patients, making use of our Translation Shortcuts. (We've also added more new Shortcut categories – including food, physical therapy, and mental health – since these browsable and searchable fixed phrases proved popular with staff members.)

5. Extended example

This section provides an example of the revised system in use. New elements introduced in the previous section are highlighted in *italics*.

Depending on the platform, the system can offer up to four input modes: speech, typing, handwriting, and touchscreen. To illustrate the use of interactive correction for speech recognition as well as machine translation, we assume that the user has clicked on the round red ***Mic Button*** to activate the microphone (Figure 1).

Still in Figure 1, notice the ***Traffic Light Icon*** and two ***Earring Icons***. These are used to switch between *Pre-check and Post-Check Modes* for translation and speech recognition, respectively. Both icons are currently green, indicating "Full speed ahead!" That is, verification has been temporarily switched off: the user has indicated that it is unnecessary to pre-check either ASR or MT before transmitting the next utterance, preferring speed to accuracy.

Just prior to the figure's snapshot, the user said, "San Jose is a pleasant city." Since verification had been switched off for both ASR and MT, these functioned without interruption. The speech recognition result appeared briefly (and in this case correctly) in the **Input Window**. Immediately thereafter the Spanish translation result (also correct in this case) appeared in the right-hand section of the **Transcript Window**, and was immediately pronounced via text-to-speech. Meanwhile, the

original English input was recorded in the left-hand section of the transcript.

Also on the English side of the transcript and just below the original English input is a specially prepared back-translation:[1] the original input was translated into Spanish, and then retranslated back into English. Proprietary techniques ensure that the back-translation means the same as the Spanish. Thus, even though *pre*-verification was bypassed for this utterance in the interest of speed, *post*-verification via the transcript was still enabled. (The **Transcript Window**, containing inputs from both English and Spanish sides and the associated back-translations, can be saved for record-keeping. *Inclusion of back-translation is new to Version 4.0.* Participant identities can optionally be masked for confidentiality.)

Using this back-translation, the user might conclude that the translation just transmitted was inadequate. In that case, or if the user simply wants to rephrase this or some previous utterance, she can click the ***Rewind Button*** (round, with chevrons). A menu of previous inputs then appears (not shown). Once a previous input is selected, it will be brought back into the **Input Window**, where it can be modified using any available input mode – voice, typing, or handwriting. In our example sentence, for instance, *pleasant* could be changed to *boring*; clicking the **Translate Button** would then trigger translation of the modified input, accompanied by a new back-translation.

In Figure 2, the user has selected the *yellow* ***Earring Icon***, specifying that the speech recognition should "proceed with caution." As a result, spoken input remains in the **Input Window** until the user explicitly orders translation. Thus there's an opportunity to make any necessary or desired corrections of the ASR results. In this case, the user has said "This morning, I received an email from my colleague Igor Boguslavsky." The name, however, has been misrecognized as "Igor bogus Lovsky." Typed or handwritten correction can fix the mistake, and the **Translate Button** can then be clicked to proceed.

Just prior to Figure 3, the ***Traffic Light Icon*** was also switched to yellow, indicating that translation (as opposed to speech recognition) should also "proceed with caution": it should be pre-checked before transmission and pronunciation. This time the user said "This is a cool program." Since the ***Earring Icon*** is still yellow, ASR results were pre-checked and approved. Then the **Translation Verification Panel** appeared, as shown in the figure. At the bottom, we see the preliminary Spanish translation, "Éste es un programa fríio." Despite the best efforts of the translation program to determine the intended meaning in context, "cool" has been mistranslated – as shown by the back-translation, "This is a cold program."

Another indication of the error appears in the **Meaning Cues Window** (third from the top), which indicates the meaning of each input word or expression as currently understood by the MT engine. Converser 4.0 employs synonyms as Meaning Cues. (In the future, pictures, definitions, and examples may also be used.) In the present case, we see that the word "cool" as been wrongly translated as "cold, fresh, chilly, …".

[1] Proprietary, and branded as Reliable Retranslation™.

To rectify the problem, the user double clicks on the offending word or expression. The **Change Meaning Window** then appears (Figure 4), with a list of all available meanings for the relevant expression. Here the third meaning for "cool" is "great, fun, tremendous, …". When this meaning has been selected, the entire input is retranslated. This time the Spanish translation will be "Es un programa estupendo" and the translation back into English is "Is an awesome program." The user may accept this rendering, despite the minor grammatical error, or may decide to try again.

The new ***Traffic Light*** and ***Earring Icons*** help to balance a conversation's reliability with its speed. Reliability is indispensible for serious applications like healthcare, but some time is required to interactively enhance it. The icons let users proceed carefully when accuracy is paramount or a misunderstanding must be resolved, but more quickly when throughput is judged more important. This flexibility, we anticipate, will be useful in future applications featuring automatic detection of start-of-speech: in Green Light Mode, ASR and translation will proceed automatically without start or end signals and thus without demanding the user's attention, but can be interrupted for interactive verification or correction as appropriate. Currently, in the same mode, for inputs of typical length (ten words or less), the time from end of input speech to start of translation pronunciation is normally less than five seconds on a 2.30 GHz Windows 7 desktop with 4.00 GB RAM, and faster in a pending cloud-based version.

6. Conclusions

Following on earlier descriptions of Converser for Healthcare, Version 3.0, and a substantial pilot project which tested it at a leading San Francisco hospital, this historical paper has conveyed hitherto unpublished details of the resulting evaluation, as presented in the healthcare organization's internal reports, based in part upon interviews carried out by an independent third-party. We have also given an account of the system revisions in Version 4.0 which resulted from this feedback and from lessons learned independently.

We expect to release Version 4.0 in early 2016, and look forward to reporting the results.

7. Acknowledgements

The authors thank the many participants in the development of Converser for Healthcare and look forward to thanking by name the organization which sponsored the pilot project for Converser discussed herein.

8. References

[1] Mike Dillinger and Mark Seligman. 2004a. "System Description: A Highly Interactive Speech-to-speech Translation System." Association for Machine Translation in the Americas (AMTA-04). Washington, DC, September 28 – October 2, 2004.

[2] Mike Dillinger and Mark Seligman. 2004b. "A highly interactive speech-to-speech translation system." In *Proceedings of the VI Conference of the Association for Machine Translation in the Americas*. Washington, D.C., September-October, 2004.

[3] Yuqing Gao, Gu Liang, Bowen Zhou, Ruhi Sarikaya, Mohamed Afify, Hong-Kwang Kuo, Wei-zhong Zhu, Yonggang Deng, Charles Prosser, Wei Zhang, and Laurent Besacier. 2006. "IBM MASTOR system: multilingual automatic speech-to-speech translator." In *HLT-NAACL 2006: Proceedings of the Workshop on Medical Speech Translation*. New York, NY, June, 2006.

[4] Mark Seligman and Mike Dillinger. 2013. "Automatic Speech Translation for Healthcare:: Some Internet and Interface Aspects." TIA (Terminology and Artificial Intelligence) 2013: Proceedings of the Workshop on Optimizing Understanding in Multilingual Hospital Encounters. Paris, France, October 30, 2013.

[5] Mark Seligman and Mike Dillinger. 2012. "Spoken Language Translation: Three Business Opportunities." Association for Machine Translation in the Americas (AMTA-12). San Diego, CA, October 28 – November 1, 2012.

[6] Mark Seligman and Mike Dillinger. 2011. "Real-time Multi-media Translation for Healthcare: a Usability Study." Proceedings of the 13th Machine Translation Summit. Xiamen, China, September 19-23, 2011.

[7] Mark Seligman and Mike Dillinger. 2008. "Rapid Portability among Domains in an Interactive Spoken Language Translation System." *COLING 2008: Proceedings of the Workshop on Speech Processing for Safety Critical Translation and Pervasive Applications*. Manchester, UK, August 23, 2008, pages 40-47.

[8] Mark Seligman and Mike Dillinger. 2006a. "Usability Issues in an Interactive Speech-to-Speech Translation System for Healthcare." HLT/NAACL-06: Proceedings of the Workshop on Medical Speech Translation. NYC, NY, June 9, 2006.

[9] Mark Seligman and Mike Dillinger. 2006b. "Converser: Highly Interactive Speech-to-speech Translation for Healthcare." HLT/NAACL-06: Proceedings of the Workshop on Medical Speech Translation. NYC, NY, June 9, 2006.

below.

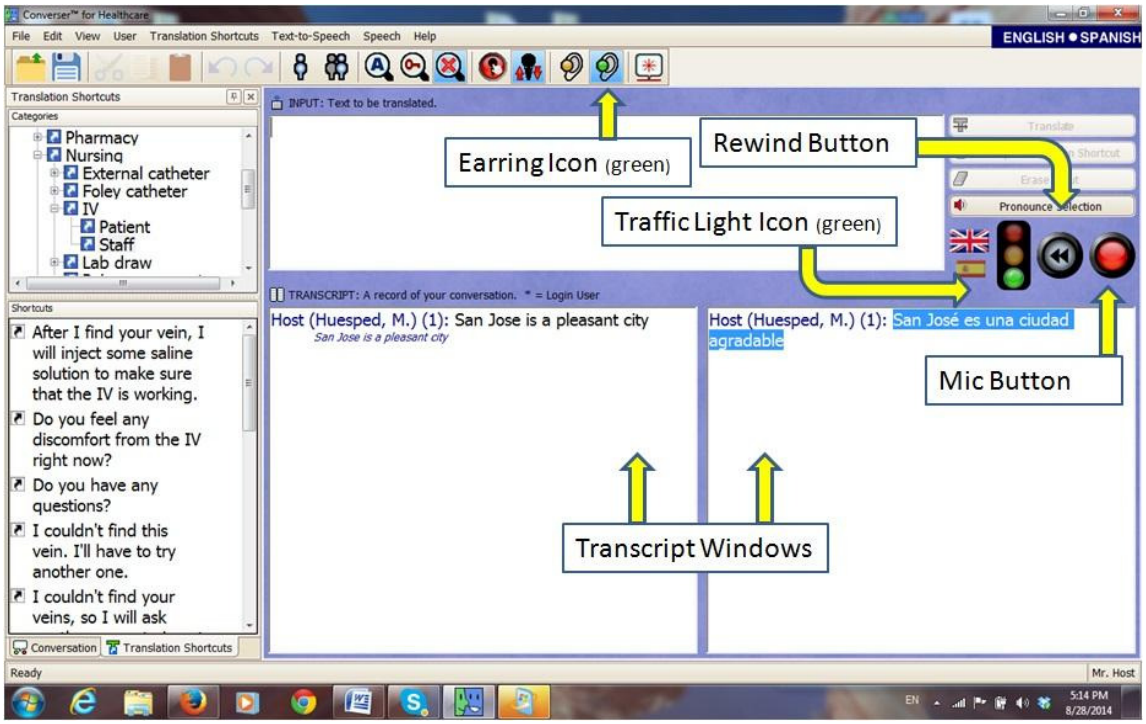


Figure 1: Earring and Traffic Light Icons are green: "Full speed ahead!"

Figure 2: Earring Icon is yellow: "Proceed with caution!"

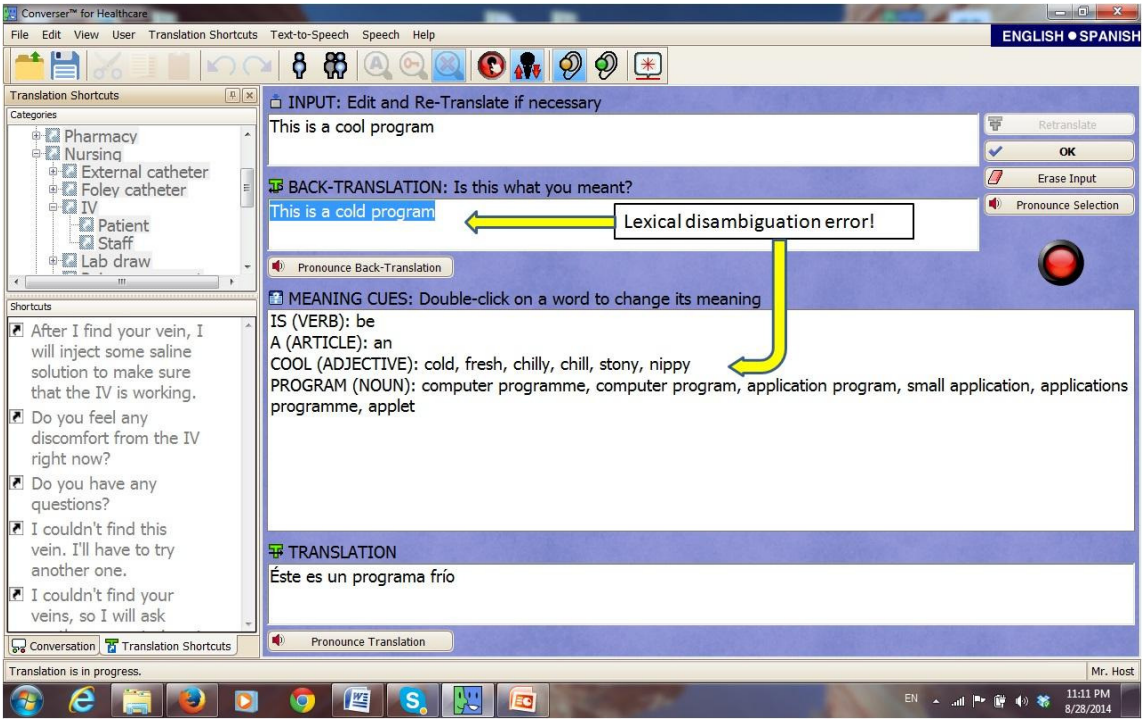


Figure 3: Verification Panel, with a lexical disambiguation error in *This is a cool program*.

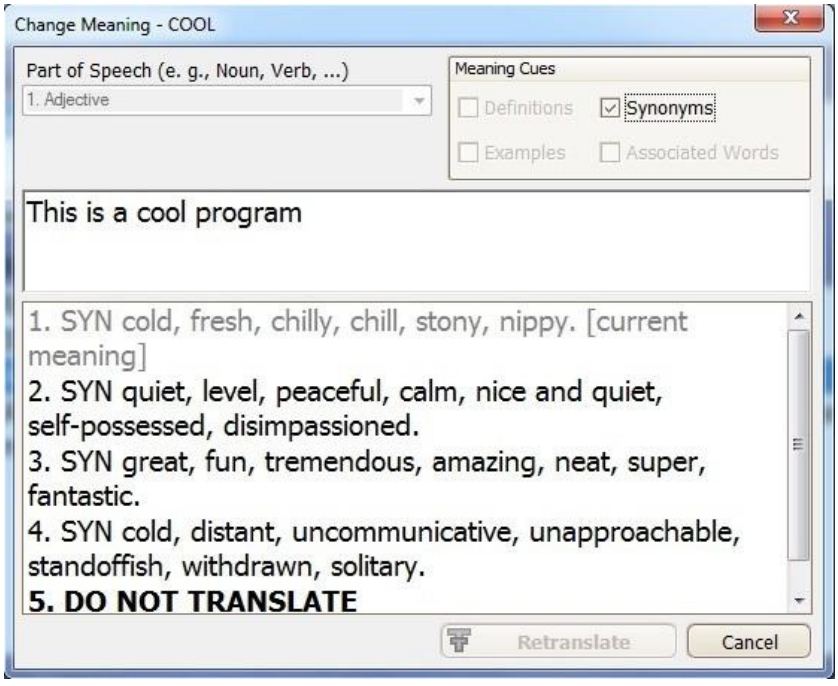


Figure 4: The Change Meaning Window, with four meanings of *cool*

Learning Segmentations that Balance Latency versus Quality in Spoken Language Translation

Hassan S. Shavarani Maryam Siahbani Ramtin M. Seraj Anoop Sarkar

School of Computing Science
Simon Fraser University, Burnaby BC. Canada
{sshavara,msiahban,rmehdiza,anoop}@cs.sfu.ca

Abstract

Segmentation of the incoming speech stream and translating segments incrementally is a commonly used technique that improves latency in spoken language translation. Previous work (Oda et al. 2014) [1] has explored creating training data for segmentation by finding segments that maximize translation quality with a user-defined bound on segment length. In this work, we provide a new algorithm, using Pareto-optimality, for finding good segment boundaries that can balance the trade-off between latency versus translation quality. We compare against the state-of-the-art greedy algorithm from (Oda et al. 2014) [1]. Our experimental results show that we can improve latency by up to 12% without harming the BLEU score for the same average segment length. Another benefit is that for any segment size, Pareto-optimal segments maximize latency and translation quality.

1. Introduction

Minimizing latency is a challenge for any spoken language translation system that does simultaneous translation. Ideally the system should produce the translation of an utterance soon after it has been produced. However, translation often involves reordering and this means that a monotone translation which immediately translates as soon as possible can be quite poor in translation quality. Waiting until the end of the input can typically improve the quality of translation but has very bad latency, while translating short segments improves latency but typically makes the quality of translation much worse. A common technique in the literature is to segment the incoming speech stream into chunks that can capture reordering between source and target languages and translate these chunks in order to improve latency.

The technique of segmenting the input is often referred to as the "salami technique" in the field of conference interpreting (by humans) [2] referring to the slicing up of the input into small, predictably sized units for translation. In spoken language translation, the "salami technique" has been mostly focused on fixed length segments or segments based on monolingual features in the input such as pauses and other similar cues [3, 4, 5, 6] to break the input into segments for incremental translation. In order to train a segmentation classifier, one can go beyond simple cues such as pauses and annotate training data with good segmentation boundaries [7, 8]. These techniques require either heuristic or human annotation of segment boundaries for some data in the source language. The segmentation classifier can be tightly integrated into a stream decoding process for incremental translation [9]. The impact of the choice of segment length has been studied in some previous work on segmentation [10] and stream decoding [11]. However, none of these approaches explicitly consider the impact of selecting between different segments (perhaps of the same size) on the translation quality in the target language. In the context of this paper, we want to choose segments that are optimal in some way with respect to latency and/or translation quality and we wish to train a segmentation strategy that provides such an optimality guarantee (on the training data).

Oda et al. (2014) [1] have explored finding segments that maximize translation quality with a user-defined bound on segment length. The training data set required for this is much more complex because, in order to optimize for segments with good translation quality, we need a training set translated with all possible segment choices and sizes and the eventual translation quality for each possible segmentation choice. Once such a training data set is built, one can apply the algorithms in [1] to find segmentation decisions that are optimal with respect to some evaluation measure of translation quality such as BLEU [12] score.

In this work, we extend previous work [1] on finding optimal segments and provide a more appealing algorithm, using Pareto-optimality, for finding good segment boundaries that can balance the trade-off between latency and translation quality. Latency is measured in terms of segments translated per second and translation quality is measured using a translation evaluation measure such as BLEU score. Using data that was produced by simultaneous translation by human interpreters, the study in Mieno et al. [13] considers how humans view the tradeoff between latency and translation quality. What they found was that humans were very sensitive to translation quality, and this implies that we need algorithms that can make a careful choice between different segmentation decisions of the same latency to produce translations with the best translation quality possible (for that latency).

In this paper we provide efficient algorithms to find segmentation decisions that explicitly rank these decisions based on the trade-off between latency and translation quality.

We provide experimental results to evaluate our approach on the English-German TED talk translation task which uses data from the IWSLT shared task data from 2013, 2012 and 2010. The results show that we can provide qualitatively better segments (compared to previous work) that improve latency without substantially hurting translation quality.

2. Segments that Maximize Translation Quality

Greedy segmentation (Oda et al. 2014) [1] is the state-of-the-art method for creating segmentation training data. In this approach, the best possible segmentation points are found over an unsegmented corpus which maximize the translation accuracy of the segmented sentences in a greedy way.

The algorithm in [1][1] has a parameter for the number of expected segments, K, which is given by Equation 1. Using this equation, the segmentation model is trained on a parallel corpus $\mathcal{F} = \langle F, E \rangle$ which has N source/target sentence pairs. $|f|$ provides the length of sentence f in words and μ is the average segment length.

$$K := max(0, \left\lfloor \frac{\sum_{f \in F} |f|}{\mu} \right\rfloor - N) \quad (1)$$

Finding each of these K segmentation points in the algorithm involves searching through all the N sentences in the corpus and examining each segment boundary in the whole corpus. For $K = 1$, one sentence in the corpus is segmented into two chunks. This way, they will produce all possible hypothesized segmentations of the entire corpus, one of which is going to be the optimal one.

Given an MT system, $\mathcal{D}$, which is already tuned on a given development set, $\mathcal{D}(f, s)$ is the translation output of the MT system $\mathcal{D}$ for a given source sentence f obtained by concatenating the translations of the individual segments defined by the set of segmentation decisions s. This set s is created by adding a segmentation point at each place where a segmentation classifier fires. In [1] the segmentation classifier is determined by checking a single feature firing. This single feature is a bigram part of speech (POS) tag. Each segmentation of the corpus is a collection of such features called Φ. Thus, s, the set of segmentation points is proportional to the number of sentences in $\mathcal{F}$ and the features Φ that determine the segments: $s \propto \{\mathcal{F}, \Phi\}$.

The accuracy score of each possible segmentation choice for a given number of segments s is computed for the whole corpus as follows:

$$B(s) = \sum_{j=1}^{N} \beta(\mathcal{D}(f_j, s), e_j) \quad (2)$$

where $\mathcal{D}(f_j, s)$ produces target translations for each source sentence f_j based on the segments in s. Each output sentence is scored by β which can be any automatic evaluation measure for translation quality. We use per-sentence smoothed BLEU score (BLEU+1) [12, 14] in this paper. $B(s)$ is the sum of the translation quality scores for each segmented sentence. The *argmax* of $B(s)$ finds the optimal segmentation for the entire corpus, searching over all possible s segment boundary points. This *argmax* of $B(s)$ is repeatedly computed for every segmentation set of size $k = 1 \ldots K$, and the set of size K is returned.

Because such an approach is computationally complex, Oda et al. (2014) [1] introduce the idea of feature grouping. Using feature grouping, once a feature has been greedily chosen, all the points exhibiting that feature are segmented at the same time and added to the set of selected features. Moreover, they take advantage of dynamic programming (DP) implementation of the greedy approach to reflect optimal feature grouping. DP is used to build larger sets of segmentation points from smaller sets. This method is called Greedy-DP or the GDP Segmentation approach in their paper.

Finally, they introduce a regularizer coefficient α to their accuracy scoring function which is aimed to control the number of selected features out of the set Φ; as a higher α will choose a smaller set of features in Φ which occur frequently to produce the necessary number of segments while a lower α tends to prefer a larger set of features in Φ, each of which occur less frequently.

$$B_\alpha(s) = B(s) - \alpha|\Phi| \quad (3)$$

In an English-German translation task, consider the three-sentence sample example of Figure 1 and the features used for choosing the segmentation points to be the bigram part of speech (POS) tags (like [1]). In this example, each point has been labeled with a general POS tag out of the set $\mathcal{P}$={N[noun], V[verb], D[determiner], J[adjective], P[preposition], S[possessive pronoun], A[adverb], R[particle], .[dot]}.

(1) I am a contemporary artist with a bit of an unexpected background .
N V D J N P D N P D J N .

(2) I was in my twenties before I ever went to an art museum .
N V P S N P N A V P D N N .

(3) I grew up in the middle of nowhere on a dirt road in rural Arkansas .
N V R P D N P N P D N N P J N .

Figure 1: Example training set for segmentation choices containing the source sentences and part of speech tags (target German sentences are not shown in this figure but appear later).

Table 1 shows the feature frequencies of the sample corpus. For $\mu = 1$ (setting each word as one segment) for the

[1]It might seem so, but we are not duplicating a lot of content from their paper, and what is included is necessary to understand our proposed algorithm. We provide an example that is used to explain our algorithm as well and which will help the reader understand the difference with our proposed algorithm. We also change their notation to match our own.

	$\Phi_{sent\ 2}$	#segments	Segmented Sentence & Translation	GDP Accuracy	PO Accuracy	Time	Segs/Sec
1	$\emptyset$	1	[I was in my twenties before I ever went to an art museum .] Ich war in meinen zwanzig vor Ich in ein kunstmuseum ging .	0.224	0.224	16.097	0.062
2	P-S	2	[I was in][my twenties before I ever went to an art museum .] Ich war in meine zwanziger vor Ich in ein kunstmuseum ging .	0.382	0.191	15.206	0.131
3	S-N	2	[I was in my][twenties before I ever went to an art museum .] Ich war in meinem zwanziger vor Ich in ein kunstmuseum ging .	0.235	0.117	15.487	0.129
4	A-V	2	[I was in my twenties before I ever][went to an art museum .] Ich war in meinen zwanzig Ich je vor ging zu einer kunst museum .	0.134	0.067	9.983	0.200
5	N-A	2	[I was in my twenties before I][ever went to an art museum .] Ich war in meinen zwanzig Ich vor in ein kunstmuseum ging .	0.224	0.112	3.462	0.577
6	N-N	2	[I was in my twenties before I ever went to an art][museum .] Ich war in meinen zwanzig vor Ich jemals zu einer kunst museum .	0.138	0.069	3.426	0.583
7	P-N	2	[I was in my twenties before][I ever went to an art museum .] Ich war in meinen zwanzig vor Ich in ein kunstmuseum ging .	0.224	0.112	2.697	0.741
8	P-S,S-N	3	[I was in][my][twenties before I ever went to an art museum .] Ich war in meine zwanziger vor Ich in ein kunstmuseum ging .	0.382	0.127	2.586	1.160
9	P-S,A-V	3	[I was in][my twenties before I ever][went to an art museum .] Ich war in meine zwanziger vor Ich je ging zu einer kunst museum .	0.272	0.090	3.137	0.956
10	P-S,N-A	3	[I was in][my twenties before I][ever went to an art museum .] Ich war in meine zwanziger vor Ich in ein kunstmuseum ging .	0.382	0.127	5.350	0.560
11	S-N,A-V	3	[I was in my][twenties before I ever][went to an art museum .] Ich war in meinem zwanziger vor Ich je ging zu einer kunst museum .	0.141	0.047	2.762	1.086
12	S-N,N-A	3	[I was in my][twenties before I][ever went to an art museum .] Ich war in meinem zwanziger vor Ich in ein kunstmuseum ging .	0.235	0.078	2.586	1.160
13	N-A,A-V	3	[I was in my twenties before I][ever][went to an art museum .] Ich war in meinen zwanzig Ich vor je ging zu einer kunst museum .	0.134	0.044	2.632	1.139

Table 2: For the second sentence in Figure 1, we show the bigram part of speech features that pick the segment boundaries, the number of segments in this sentence, the accuracy for both the Greedy-DP (GDP) algorithm of [1] and our Pareto-Optimal (PO) algorithm (see Section 3), the translation times and latency measurements (with parameter $\mu = 8$). GDP accuracy is different from PO accuracy because accuracy is measured differently in the two approaches.

Feat	Freq	Feat	Freq	Feat	Freq
N-P	6	J-N	3	V-R	1
P-D	5	N-N	2	P-S	1
D-N	4	P-N	2	P-J	1
N-.	3	D-J	2	S-N	1
N-V	3	R-P	1	A-V	1
V-D	3	N-A	1		
FSS Size				40	

Table 1: Frequencies of the bigram part of speech tags in the example from Figure 1.

example in Figure 1, the GDP segmentation algorithm will set $K = 40 = max(0, \left\lfloor \frac{[\sum_{f \in F} |f|=43]}{[\mu=1]} \right\rfloor - [N = 3])$. Likewise, if we set $\mu = 8$, we will have $K = 2$, and our possible segmentation sets will be in {{N-N}, {P-N}, {D-J}, {[R-P],[N-A]}, {[V-R],[P-S]}, ...} for our running example. Therefore, the segmentation set will contain all the different ways to segment the segmentation training data to obtain the average segment length of 8. If we want to consider different possible segmentations of the second sentence in our sample corpus with $\mu = 8$, the possible segmentations will be one of the sets inside $s_{possible}$.

$s_{possible}$={{}, {N-N}, {P-N}, {{N-A}, {P-S}}, {{N-A}, {S-N}}, {{A-V}, {P-S}}, {{A-V}, {S-N}}, {{A-V}, {N-A}}, {{P-S}, {S-N}}}.

Table 2 shows the possible segmentations of the second sentence of the example in Figure 1 for $K = 2$. We show Φ only for the second sentence, so when $\Phi_{sent\ 2}$ is $\emptyset$ the two segments were chosen in other sentences not shown in this table. The GDP algorithm will choose the segmentation that maximizes accuracy, so for $K = 2$, the GDP algorithm will pick either sentence 8 or 10 from Table 2 (the algorithm has to break ties arbitrarily in the sorted order for segmentations with equal accuracy).

The GDP algorithm thus picks the segmentation decisions that result in the best accuracy on the training set. However, the GDP algorithm considers only accuracy to find the optimal segmentations, so it tends to prefer larger segments that can result in worsening the latency. Furthermore, the trade-off between accuracy and latency is not modelled in the search for good segmentations. This trade-off is crucial in the design of simultaneous translation systems. Another issue can be observed in Table 2, in choosing to spread the segmentation points to more sentences or concentrating them in fewer sentences, the GDP algorithm tends to choose the latter in spite of the regularizer on the size of Φ. Equation 3 does not consider the number of segments which are placed in each individual sentence. We try to address both of these issues in our Pareto-optimal segmentation approach.

3. Pareto-Optimal Segmentation Approach

In this section, we will show how Pareto-optimality can help producing a better segmentation with respect to both latency and accuracy. To get to this point, we will first review the concept of Pareto-optimality as it shows how one could choose different equally important points in the two-dimensional space of latency and accuracy.

Considering translation latency-accuracy points depicted in Figure 2 as an example, a point will be *Pareto-Optimal if*

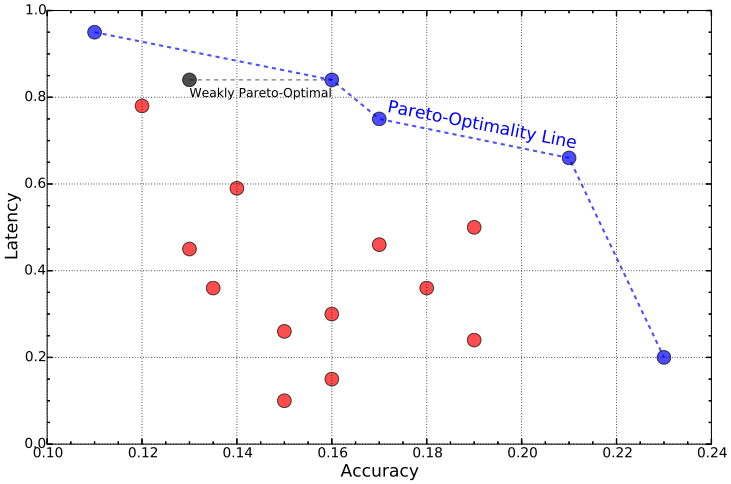


Figure 2: Pareto-Optimal and Weakly Pareto-Optimal points as well as the dominated points scored on the two metrics of interest in this paper: latency and translation accuracy scores (e.g. BLEU*).*

and only if there is no other point which is both faster and more accurate than this point (or even equal in one aspect). In other words, a point p_1 is Pareto-optimal if and only if for each point p_2 in the region we have

$$\Lambda\{p_2\} < \Lambda\{p_1\} \;\&\; B\{p_2\} < B\{p_1\} \tag{4}$$

where Λ and B are representing functions measuring latency and accuracy. Therefore, point p_1 dominates any such point p_2, shown as $p_1 \rhd p_2$. If the dominated point p_2 has an equal latency or accuracy measure to the dominating point p_1, we call p_2 a Weakly Pareto-Optimal point.

Based on these concepts and paying attention to the Pareto-Optimality Line in Figure 2, we see that there may be more than one optimal point on which one could tune the MT system to enhance the performance of stream decoding. Each of these points is one *Pareto Frontier Point*. The Pareto frontiers provide a range of equally optimal points rather than one most accurate point and we use this fact in our search for optimal segments.

In our approach, we use the same notation of K and μ introduced in the Greedy approach of Section 2 to explore the space of possible segmentations in the training corpus. However, in our algorithm, the parameter μ (the average segment length) can be seen as a way to explore the trade-off between latency and accuracy. Longer segments (with a higher μ value) tend to be associated with higher translation quality. But the cost of this higher accuracy is that our translation system will have a worse latency. Shorter segments (with a smaller μ value) tend to be associated with better latency (on average there will be more segments translated per second). In this case the translation fluency scores tend to become worse. We compare our approach to the Greedy approach by (Oda et al., 2014) [1] which takes the value of K as an input. We consider different values of K in our algorithm to balance the latency-accuracy trade-off.

We search for the best set s, containing K segments (total number of expected chunks) over the stream of an expected known size. The cardinality of this segmentation set may vary from 0 (no segmentation at all), to $W = \sum_{i=1}^{N}\{|f_i|-1\}$ (take each word as a segment). A *'full segmentation set'* (*FSS*) will contain all possible W segments. $\mathcal{S}_{all}$ represents a superset containing all possible segmentation sets over F (source sentences in parallel corpora).

$\mathcal{S}^* \in \mathcal{S}_{all}$ is defined as a set of best segmentation strategies which maximizes an evaluation function over latency and accuracy (Equation 7). We propose two scoring functions for latency and accuracy (Equations 5 and 6 respectively) which are used in Equation 7.

We modify the accuracy function of Equation 3 to address the problem of spreading the segmentation positions (Equation 5).

$$B_\alpha(s) = \sum_{j=1}^{N} \frac{\beta(\mathcal{D}(f_j, s_j), e_j)}{|s_j|} - \alpha|\Phi| \tag{5}$$

where $K = |s| = \sum_{j=1}^{N} |s_j|$ holds and $|s_j|$ is the number of segments (i.e. the number of segmentation points plus one) for each sentence f_j.

The latency scoring function is defined as the average number of segments translated in the unit of time, which can be simply computed by dividing the the total number of segments by the total translation time as follows:

$$\Lambda_\alpha(s) = \frac{|s|}{\sum_{j=1}^{N} \gamma(\mathcal{D}(f_j, s))} - \alpha|\Phi| \tag{6}$$

where γ function measures the time taken for computing $\mathcal{D}(f_j, s)$. Note that, here we use the same regularization strategy used in Equation 3 (see Section 2).

$$\mathcal{S}^* = \underset{s\in\mathcal{S}_{all}}{\arg \text{pareto frontier}} \{B_\alpha(s), \Lambda_\alpha(s)\} \tag{7}$$

Note that in Equation 7, the output of *"arg pareto frontier"* is the Pareto-optimality line in the accuracy-latency plot. Therefore, $\mathcal{S}^*$ might contain more than one best set of segmentations.

$\mathcal{S}^*$ can be found using a naïve algorithm as described in Algorithm 1. However, this algorithm is computationally expensive and its time complexity is exponentially increased by increasing the size of K.

Algorithm 1 Pareto-Optimal Segmentation

1: $\mathcal{S}_0^* \leftarrow \emptyset$
2: **for** $k = 1$ to K **do**
3:

$$\mathcal{S}_k^* \leftarrow \underset{p\in FSS \wedge p\notin\mathcal{S}_{k-1}^*}{\arg \text{pareto frontier}} \left\{ \begin{array}{c} B_\alpha(\mathcal{S}_{k-1}^* \cup \{p\}), \\ \Lambda_\alpha(\mathcal{S}_{k-1}^* \cup \{p\}) \end{array} \right\}$$

4: **end for**
5: **return** $\mathcal{S}_K^*$

Algorithm 2 depicts our *Computationally Efficient Pareto-Optimal Segmentation Method* to find $\mathcal{S}^*$. The main

Algorithm 2 Computationally Efficient Pareto-Optimal Segmentation

1: $\Phi_0 \leftarrow \emptyset$
2: **for** $k = 1$ to K **do**
3: **for** $j = 0$ to $k-1$ **do**
4: $\Phi' \leftarrow \{\phi : (\phi \notin \Phi_j) \wedge (count(\phi; \mathcal{F}) = k - j)\}$
5: $\Phi_{k,j} \leftarrow \Phi_j \cup \left\{ \arg \text{pareto frontier}_{\phi \in \Phi'} \{B_\alpha(s(\mathcal{F}, \Phi_j \cup \{\phi\})), \Lambda_\alpha(s(\mathcal{F}, \Phi_j \cup \{\phi\}))\} \right\}$
6: **end for**
7: **if** $k < K$ **then** ▷ *To reduce the computational complexity*
8: $\Phi_{k,j} \leftarrow \text{argmax}_{\phi \in \{\Phi_{k,j} : 0 \le j \le k\}} B_\alpha(s(\mathcal{F}, \phi))$
9: **end if**
10: $\Phi_k \leftarrow \arg \text{pareto frontier}_{\Phi \in \{\Phi_{k,j} : 0 \le j \le k\}} \{B_\alpha(s(\mathcal{F}, \Phi)), \Lambda_\alpha(s(\mathcal{F}, \Phi))\}$
11: **end for**
12: **return** $s(\mathcal{F}, \Phi_K)$

loop (lines 2-11) each time finds the next best segmentation feature (ϕ) and adds it to the set of best segmentation points which are already found (creating a set of k points). Each feature is a bigram part of speech tag. The inner loop (line 3) implements the dynamic programming (DP) condition as in [1]. For instance, for Φ_3 this inner loop would combine the features in the set Φ_0, Φ_1 and Φ_2 (for $j = 0, 1, 2$) with the features that occur with a count of $3, 2$ and 1 respectively. So take $\Phi_{3,1}$ which is the set that is updated in line 5, the points satisfying the Pareto frontier criteria are selected out of Φ' and combined with the segmentation points of the chunked sub-segments. $\Phi_{3,1}$ contains the union of all features in Φ_1 computed previously in the DP table with new features of count 2 collected in line 4. Eventually $\Phi_{3,1}$ is used to search over Pareto frontier candidates to produce Φ_3 in line 10. Line 7 limits the computational complexity of producing Pareto frontiers out of a set containing previously computed Pareto frontiers. This line sets the most accurate point out of currently discovered Pareto frontiers, to be the only Φ of the next step (to build Φ_{j+1}). In Line 10, all possible segmentation points are analyzed (for $k < K$ there is just one point) and the Pareto frontiers out of them are stored as Φ_k. Finally, in line 12, the result of segmentation with the discovered segmentation points of Φ_K is produced and returned.

Performing the same segmentation task from Section 2, over our running example using this Pareto-optimal segmentation approach, will initially result in the same segmented sentences but our algorithm has a different intuition about choosing the best translations. Table 2 reports PO segmenter accuracy, total translation time and latency measurement values besides the reported accuracy of GDP segmenter over different segmented versions of second sample example (in Figure 1) as well as the actual feature set (Φ) for each specific segmentation and translation.

To explain the algorithm we have used the running example in Figure 1 and traced the output of our Algorithm 2 for $K = 2$ and provided the plot of accuracy-latency values during one execution of this algorithm in Figure 3. We get the highest accuracy with the worst latency in the beginning. Then the algorithm starts to find the first best segmentation point ($K = 1; j = 0$) and it finds four possible

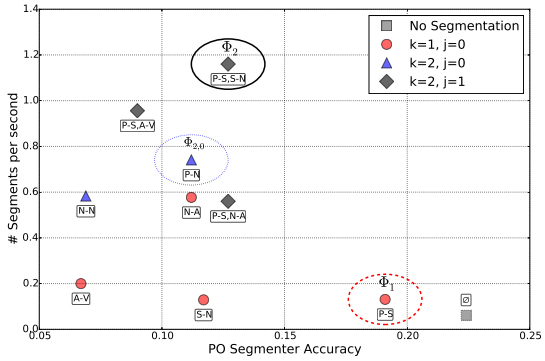


Figure 3: Evaluation results of different segmentation strategies in one loop of the algorithm 2

segmentation candidates (depicted as filled circles in Figure 3). It chooses the best accuracy as Φ_1 and moves to the next round to find the second (last) segmentation point. It first considers features happening twice ($K = 2; j = 0$), then it again chooses the best accuracy as $\Phi_{2,0}$. Next, it examines the strategy of adding a single repeated feature to Φ_1 which ends up to the points depicted as diamonds. When it finds the second strategy which dominates the first strategy, it chooses the Pareto-optimal points out of the new strategy and reports it as Φ_2. Although in this example, the final segmentation set (Φ_2) contains just one point, this is not always the case.

4. Experiments

4.1. Experimental Setup

We evaluate our approach on the English-German TED speech translation data [15]. We used Moses [16] which is a conventional phrase-based SMT system using the standard set of features in the discriminative log-linear model for SMT to produce the translations for each possible segmentation decision in our segmentation training data. We used the Stanford POS-Tagger [17] to tokenize and produce the POS tags over the train and test data. We used *IWSLT 2013 Train data* plus half of the *Europarl data* [18] to train our MT system on English-German and *IWSLT Test 2012* to tune it using MERT [19]. Our German language model was trained using

the *monolingual data from WMT 2013 Shared Task* [2]. The segmentation training data was taken from IWSLT Shared Task *Dev 2010 and 2012 and Test 2010* and it has been tested on IWSLT Shared Task *Test 2013*. Table 3 shows the statistics of data used in our experiments.

	Sentences	Types	Tokens
MT Train	1033491	105267	27948041
MT Tune	1730	3937	31568
Seg Train	3669	6773	74883
Seg Test	1025	3181	22026

Table 3: Size of datasets used in our experiments.

For the evaluation metrics used to evaluate segment translation quality and latency, we use BLEU+1 [12, 14] and the number of translated segments per time unit (S/T), respectively. We set the α regularizer coefficient to 0.5, for both GDP and Pareto-Optimal (PO) segmenters. This value for α avoids selecting features with extremely high or low frequency.

We train the MT system and use it in all experiments. Using the trained MT system, we translate all possible segments and store them in a lattice (like [1]). In this way, we can access to a translation instantly while computing the evaluation metrics (Equation 7).

We compute the time of translations over each segment in order to evaluate the latency of translations. However, this computed translation time is the result of many factors and different seek and search algorithms and may depend on low-level issues such as cache misses on the hardware where the MT system is running. Our results were consistent across many runs so we do not consider such issues to be dominant in our experimental results.

4.2. Accuracy vs. Latency-Accuracy Evaluation

In this experiment, we would like to assess the effect of adding latency to accuracy metric in the segmentation task. In our experiments, we use two baselines: the state-of-the-art speech segmenters (Rangarajan et al. 2013) [5] and GDP (Oda et al. 2014) [1]. We implemented a heuristic segmenter based on (Rangarajan et al. 2013) [5] which segments on surface clues such as punctuation marks. These segments reflect the idea of segmentation on silence frames of around 100ms in the ASR output used in [4]. This type of heuristic segmenter is a special case of a PO segmenter which inserts segment boundaries only for POS bigrams that end with a punctuation POS tag.

We ran our PO segmenter and the GDP segmenter with different values of parameter μ (average segment length) between 2 and 15 as well as the heuristic segmenter over the same data explained in Section 4.1. Due to the large number of generated points and outputs, we summarize the results in Figure 4 and Figure 5.

[2] http://statmt.org/wmt13/translation-task.html

Our experiments show that different possible values of μ will divide the accuracy-latency area into districts and each experiment is expected to exhibit a number of samples of each district for each μ. We show each district with a circle in the figures as the representative of the group of obtained points relating to one specific μ. This circle is put in the centroid of the points in the group. To show the size ratio of districts to each other, the more points found in one district, the bigger the circle is depiceted. But not all the points in the group are Pareto-optimal, so we add another circle inside the outer one showing the ratio of Pareto-optimal points to the whole group of points. If all of the points for one μ were Pareto-optimal, both circles would have the same radius and the inner circle would not have been visible. In addition, we show the results of the baseline heuristic and GDP segmenter using $\Diamond$s and *X*s, respectively. Moreover, we plot the real Pareto-optimality line with the actual points on it to give the reader the chance to compare the actual results of the experiments to the baseline results.

Our choice of the axis is different from previous work in this area. Commonly, segmentation results are reported with accuracy on the y-axis, but we use the x-axis instead in order to easily get a better visual understanding of pushing the Pareto-optimality line towards the trade-off area we care about (the "knee" of our plots).

Figure 4 shows the latency (average number of segments translated per second) and translation quality (BLEU) on the training data. Figure 5 shows the latency and accuracy on the unseen test set. These figures show that Pareto-optimality is a useful methodology to explore the various options for segmentation boundary selection. Optimizing for Pareto-optimality leads to segmentations that provide latency and accuracy improvements simultaneously and provide choices for the trade-off between latency and accuracy.

While Figure 5 shows the overall trend for various segment sizes on the test data, we chose some specific segment lengths and show a head to head comparison between the two segmenters in Table 4. This comparison shows that our PO segmenter can provide faster latency compared to the GDP segmenter while retaining translation accuracy.

	$\mu = 3$		$\mu = 8$	
	Latency	Accuracy	Latency	Accuracy
GDP	0.424	0.18	0.305	0.21
PO	0.474	0.18	0.315	0.21

Table 4: Result comparison for $\mu = 3$ and $\mu = 8$.

Our approach of optimizing over the latency in addition to the translation quality always results in better latencies compared to the baseline while keeping the same translation quality or even improving it in some cases.

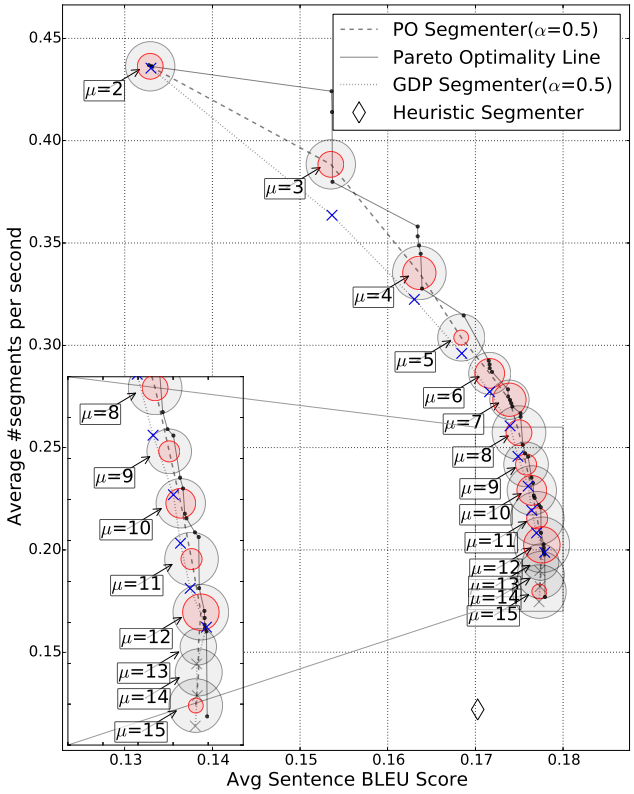


Figure 4: Comparison on the segmentation training data.

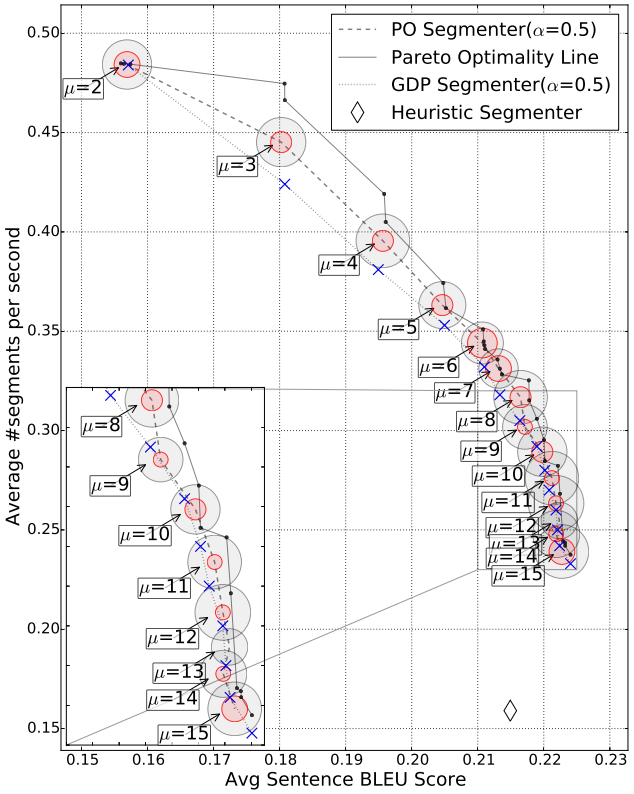


Figure 5: Comparison on the segmentation test data.

5. Related Work

In speech translation, the segmentation task can be performed on speech or the transcribed text. Early work on speech translation uses prosodic pauses detected in speech as segmentation boundaries [3, 4]. Segmentation methods applied on the transcribed text can be divided to two categories: heuristic methods which use linguistic cues, like conjunctions, commas, etc. [5]; and statistical methods which train a classifier to predict the segmentation boundaries. Some early methods use prosodic and lexical cues as features to predict soft boundaries [20]; while most recent methods rely on word alignment information to identifies contiguous blocks of text that do not contain alignments to words outside them [7, 8]. In addition to these segmentation approaches which are applied before calling the translation decoder, there is another strategy which perform the segmentation during decoding which is usually called stream or incremental decoding. Different incremental decoding approaches have been proposed for phrase-based [11, 21] and hierarchical phrase- based translation [8, 22]. He et al. [23] focus on language pairs with divergent word order by designing syntactic transformations and rewriting batch translations into more monotonic translations. Some research has been conducted on human simultaneous interpretation to determine the effect of the latency and accuracy metrics on the human evaluation of the output of simultaneous translation. The results indicate that latency is not as important as accuracy [13]. This implies that we need algorithms that can make a careful choice between different segmentation decisions of the same latency to produce translations with the best translation quality possible (for that latency) which we have done in this paper.

6. Conclusion

This paper explores multi-metric optimization in simultaneous translation that learns segmentations that optimize both latency and translation quality. We provide an efficient algorithm for Pareto-Optimal segmentation and conducted a series of experiments that compared our approach to Oda et al. [1] which used translation quality as the only criteria to select segmentation choices. We showed that Pareto-optimality provides a better trade-off between latency and translation quality. For any segment size, Pareto-optimal segments maximize latency and translation quality.

In future work, we plan to iteratively use a weighted segmentation model that is trained using the Pareto frontier in order to iteratively find new weights for the segmentation model that will extend the "knee" of the Pareto frontier. Such an approach was explored in [24] for multi-metric tuning of SMT models, but has not been explored for training a segmentation model.

7. References

[1] Y. Oda, G. Neubig, S. Sakti, T. Toda, and S. Nakamura, "Optimizing segmentation strategies for simultaneous speech translation," in *ACL*, 2014.

[2] R. Jones, *Conference Interpreting Explained*, ser. Translation Practices Explained. Taylor & Francis, 2014.

[3] C. Fügen, A. Waibel, and M. Kolss, "Simultaneous translation of lectures and speeches," *Machine Translation*, vol. 21, no. 4, pp. 209–252, 2007.

[4] S. Bangalore, V. K. Rangarajan Sridhar, P. Kolan, L. Golipour, and A. Jimenez, "Real-time incremental speech-to-speech translation of dialogs," in *Proc. of NAACL HLT 2012*, 2012, pp. 437–445.

[5] V. K. Rangarajan Sridhar, J. Chen, S. Bangalore, A. Ljolje, and R. Chengalvarayan, "Segmentation strategies for streaming speech translation," in *NAACL*, 2013.

[6] T. Fujita, G. Neubig, S. Sakti, T. Toda, and S. Nakamura, "Simple, lexicalized choice of translation timing for simultaneous speech translation." in *INTERSPEECH*, 2013, pp. 3487–3491.

[7] M. Yarmohammadi, V. K. R. Sridhar, S. Bangalore, and B. Sankaran, "Incremental segmentation and decoding strategies for simultaneous translation," in *Proc. of IJCNLP-2013*, 2013.

[8] M. Siahbani, R. Mehdizadeh Seraj, B. Sankaran, and A. Sarkar, "Incremental translation using a hierarchical phrase-based translation system," in *Proceedings of IEEE Spoken Language Technology Workshop (SLT 2014)*, 2014.

[9] M. Siahbani, B. Sankaran, and A. Sarkar, "Efficient left-to-right hierarchical phrase-based translation with improved reordering," in *Proc. of EMNLP*, Seattle, USA, October 2013.

[10] M. Wolfel, M. Kolss, F. Kraft, J. Niehues, M. Paulik, and A. Waibel, "Simultaneous machine translation of german lectures into english: Investigating research challenges for the future," in *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*. IEEE, 2008, pp. 233–236.

[11] M. Kolss, S. Vogel, and A. Waibel, "Stream decoding for simultaneous spoken language translation." in *INTERSPEECH*, 2008, pp. 2735–2738.

[12] K. Papineni, S. Roukos, T. Ward, and W. jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, 2002, pp. 311–318.

[13] T. Mieno, G. Neubig, S. Sakti, T. Toda, and S. Nakamura, "Speed or accuracy? a study in evaluation of simultaneous speech translation," in *INTERSPEECH*, 2015.

[14] D. Lin and F. Och, "Orange: a method for evaluating automatic evaluation metrics for machine translation," in *COLING 2004*, 2004, pp. 501–507.

[15] 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.

[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 *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.

[17] K. Toutanova, D. Klein, C. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of HLT-NAACL 2003*, 2003, pp. 252–259.

[18] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT Summit*, 2005.

[19] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of ACL*, 2003.

[20] E. Matusov, D. Hillard, M. Magimai-doss, D. Hakkani-tur, M. Ostendorf, and H. Ney, "Improving speech translation with automatic boundary prediction," in *In Proc. Interspeech*, 2007, pp. 2449–2452.

[21] B. Sankaran, A. Grewal, and A. Sarkar, "Incremental decoding for phrase-based statistical machine translation," in *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, ser. WMT, 2010.

[22] A. Finch, X. Wang, M. Utiyama, and E. Sumita, "Hierarchical phrase-based stream decoding," in *Proc. of EMNLP*, Lisbon, Portugal, September 2015.

[23] H. He, A. Grissom II, J. Morgan, J. Boyd-Graber, and H. Daumé III, "Syntax-based rewriting for simultaneous machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015.

[24] B. Sankaran, A. Sarkar, and K. Duh, "Multi-metric optimization using ensemble tuning," in *NAACL*, 2013.

Development of a Filipino-to-English Bidirectional Statistical Machine Translation System that dynamically updates via user feedback

Jasmine Ang, Marc Randell Chan, John Paolo Genato, Joyce Uy, Joel Ilao

College of Computer Studies
De La Salle University – Manila
jasmine_ang@dlsu.edu.ph, marc_chan@dlsu.edu.ph, paolo_genato@yahoo.com,
joyce_uy@dlsu.edu.ph, joel.ilao@delasalle.ph

Abstract

In this paper, we describe a Moses-based statistical machine translation (SMT) system, called FEBSMT, that incorporates periodic user feedback as a mechanism that allows the SMT system to adapt to prevailing translation preferences for commonly queried phrases, and assimilate new vocabulary elements in recognition of the dynamically changing nature of languages. A parallel corpus containing a total of ~22K sentences in the tourism domain was used in developing the system. Updating the SMT's language model and phrase tables via user feedback was modeled after the Post-Edit Propagation (PEPr) system [6]. Incremental training iterations were performed on the developed system via user feedback, which were collected in a duration of three months. The developed system was evaluated using the BLEU, NIST, METEOR, and TER metrics. We noted that the Filipino-to-English translations consistently scored higher than the English-to-Filipino translations. Over the course of 100 training iterations using randomly selected sentences taken from a closed set of sentences provided with user feedback, it was observed that the translation accuracy sharply improves within the first few iterations, which then gradually tapers after a peak translation performance has been reached.

1. Introduction

In the context of facilitating communications among citizens of ASEAN member countries especially as it prepares for economic integration in 2015, the ASEAN Machine Translation (ASEAN-MT) Project was launched [5]. The initial design of the ASEAN-MT system uses English as a pivot language to perform translation between pairs of major languages of the ASEAN member countries.

Furthermore, these machine translation systems can also contribute to the United Nations Millennium Goal of Developing a Global Partnership for Development [8]; since, one of its target condition is to "make available benefits of new technologies, especially information and communications" through the use of the Internet. However, not all pieces of information are available in English and not all are translated correctly. Hence, multiple improved machine translation systems are effective in developing bridges for information dissemination.

2. Related Work

2.1 Tools

2.1.1 Moses

Moses is an open-source toolkit for statistical machine translation that allows one to automatically train translation models for any language pairs [3]. It does training for any language pair with the use of a parallel corpus. The parallel corpus is separated into training, development and testing sets. The training set is where the bilingual phrases are extracted and their weights are learned. The development set is used to adjust the values of the parameters of the decoder, while the testing set is used for assessing the translation quality. For this project, Moses setting chosen for training the Filipino-English bidirectional SMT system are as follows: language model (LM) order of 3, cleaning range of 1-80, and the decoder's distortion limit of 6 [4]. The setting for FEBSMT was based from the previous Philippine Component of the ASEAN project in order to track the improvement in the machine translation technology.

2.1.2 PEPr

Post-edit Propagation (PEPr) is a phrase-based statistical machine translation system and uses an automatic post-editing (APE) setting with learning capabilities [6]. The APE system automatically post-edits the machine translation output into a proper text with human quality. Moreover, this approach aims to handle various errors, ranging from determiner selection to grammatical agreement. The APE system is built using the data comprised of the baseline translations and their post-edited counterparts.

In performing the Post-edit Propagation, the system has to undergo a cycle of two processes as shown in Figure 2.1.

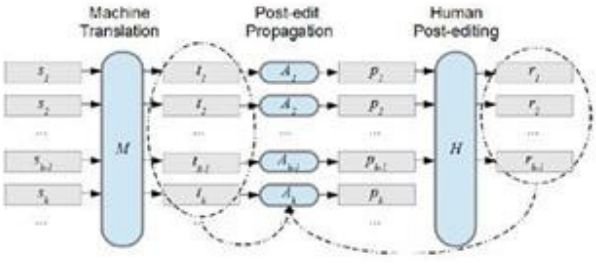


Figure 2.1 The Feedback Process of PEPr. Figure extracted from [6]

The first process involves the training of the baseline system, labeled M in Figure 2.1. The output from the baseline system is passed into the APE system. The baseline system is treated as a black box since no modification will be performed. The second process involves the APE system and a human post-editor. The baseline translations are subjected to human post-edits and these pairs of texts are used for the training of the APE system. Further version of APE systems were trained using the translations of the previous APE with their corresponding post-edits.

The APE system relies on the phrase table and language model of the previous APE version and combines them to the current. For the language model combination, linear mixture model is applied; while for the phrase table combination, linear interpolation is also applied. This process is used to broaden the vocabulary in the language model and balance the probability of the phrase table based on the post-edits.

When translating, the input text will first pass through the baseline system, and then pass through the latest APE system to be automatically post-edited. For this research, the concept of PEPr's APE system was applied due to its flexibility in taking user feedback or user post-edits as input to build the next APE system to improve machine translation quality.

3. System Design of FEBSMT

The aim of FEBSMT is to use post-editing approach to improve the translation accuracy of the machine translation. It is a web service application wherein users can translate Filipino and English bi-directionally. The development of FEBSMT is composed of three phases, namely, the training phase, the development phase, and the testing phase. The discussions of these phases are found in the succeeding sections.

3.1 Training Phase

The training phase consists of two parts. The first part of the training phase is data gathering and the second part is the data cleaning.

3.1.1 Data

The data was gathered from the Center for Language Technologies (CeLT) of De La Salle University (DLSU). It is a Filipino-to-English parallel corpus containing 22,031 sentences of a parallel corpus in the tourism domain. The parallel corpus is randomly split into 70% for the training of the baseline system (Block *M* in Figure 2.1). For evaluation, 10% of the data were selected for testing the machine translation accuracy and 20% were used for the development set.

3.1.2 Data Cleaning

Cleaning of data ensures that the data does not contain spelling errors, special characters, and tags. The data was also tokenized and re-cased into their lowercase.

3.2 Development Phase

For this phase, the development set from the data was used for the simulation of the feedback mechanism. This set of data was used to build multiple APE versions for 100 iterations. This is to observe the changes in the translation quality per APE version. For instance, the sentence "This is from room 208." is translated into "Ito mula sa room 208.". Although the translation is semantically correct, however it is grammatically incorrect. The proper translation should be "Mula ito sa room 208.". This is to be used as the feedback for the APE.

3.3 Testing Phase

A dataset containing 10% of the parallel corpus was used as the testing data for verifying the accuracy of the machine translation using the four evaluation metrics: BLEU [5], NIST [2], METEOR [1], and TER [7]. The testing data was made constant in order to provide a consistent evaluation of FEBSMT. For this project, APE was implemented and initially ran for five times on the same corpus. Every instance of the five iterations and the baseline system was subjected to the testing. The results showed a convergence of all the metric scores.

4. Results and Discussion

This section enumerates and explains the procedures of the succeeding experiments using both Filipino-to-English and English-to-Filipino sets of different human feedback and testing data. It also discusses the purpose of each experiment, along with its results. The results were analyzed and evaluated with the different evaluation metrics. Furthermore, the results will be the basis for the evaluation of the entire FEBSMT system.

For the experiments, the baseline development data and human feedback data were used in conducting the experiments with each set differing in size, content, and context. Two sets of testing data were used for testing the incremental training approach: the 10% baseline testing data and the 100 sentences randomly selected from the entire tourism corpus. This was done to maintain a consistent comparative reference to each other and to the baseline system.

For the automated evaluation metrics, four metrics were used, namely: BLEU, NIST, METEOR, and TER. BLEU and NIST are both precision-based metrics, which score the number of the target translation matches to the reference. METEOR, an F-score metric, measures precision and recall, the number of matches between the target, the reference and their explicit word ordering. TER counts the number of post-edits required to change the target translation to the reference. For the evaluation metric score of BLEU, NIST, and METEOR, a higher value means more matched words between the translation output and reference translation. If the score for TER is lower, the similarity between the translations is greater.

4.1 APE Training with Human Feedback

The purpose of this experiment is to determine if incrementally training the system using human feedback will improve the machine translation quality.

For this experiment, 20% of the tourism corpus was used as the development set. The development set was translated in the baseline system and was subjected to manual post-editing to be used as the feedback for the 100 incremental training iterations. In each incremental training phase of the APE, a total of 1000 sentences were randomly selected from the baseline translation of the development set paired with their corresponding post-edited

counterpart. For evaluating the system, 10% of the same corpus was used as testing data.

The differences between the translation quality of Filipino-to-English and English-to-Filipino in terms of their evaluation scores can be observed in Figures 4.1 to 4.4. The range of scores throughout the 100 incremental training iterations for Filipino-to-English translation is between 0.36 to 0.38, while 0.32 and 0.33 for English-to-Filipino translation. Fluctuations and abrupt increase of scores occurred for both experiments. The peak in the translation scores occurred in the 6th iteration for the Filipino-to-English translation, and in the 15th iteration for the English-to-Filipino translation, obtaining a BLEU score of 0.3795 and 0.3346, respectively.

The scores of the 6th and 15th APE iterations, which obtained the highest scores, however, still have values lower than the baseline score. This means that while there is an inconsistency in the scores of the two experiments, the baseline system still displayed a translation that is closer to a human quality base from the BLEU and NIST scores but the TER evaluation metric was higher by 0.4071 and the METEOR score was very low. These observations suggest that the baseline system's translations have many extraneous words and incorrect word reordering.

A word can have many different translations coming from different contexts, and this tendency was observed to be the possible cause of the decrease in scores. In comparison, there is a more apparent decline in the scores of Filipino-to-English, unlike in the English-to-Filipino where the scores were significantly fluctuating.

4.2 Error Analysis

For thorough comparison, the results from baseline, 6th, 15th, and 100th incremental training were selected. Baseline is necessary to serve as the benchmark of comparison. The 6th and 15th incremental training was chosen for having the highest resulting BLEU score for English-to-Filipino and Filipino-to-English, respectively. This is to observe whether the value of the BLEU score has any effect on the actual translations. Lastly, the 100th incremental training was chosen as a representative of future incremental training of the APE system.

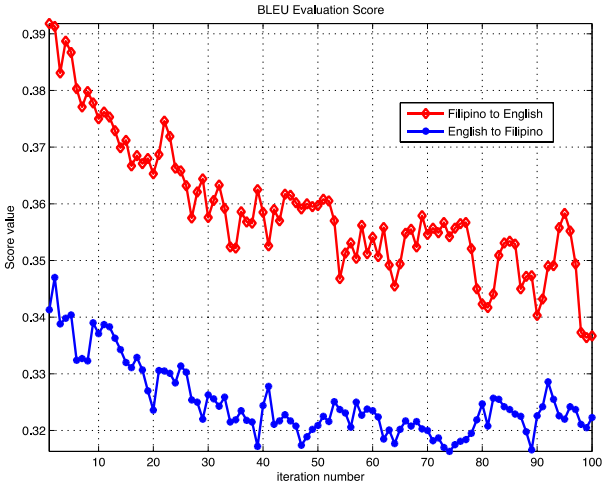


Figure 4.1: Bi-directional Filipino-to-English BLEU Evaluation Score using Human Feedback

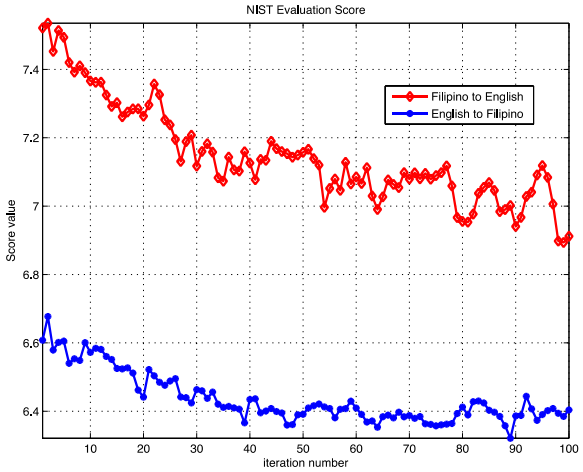


Figure 4.2: Bi-directional Filipino-to-English NIST Evaluation Score using Human Feedback

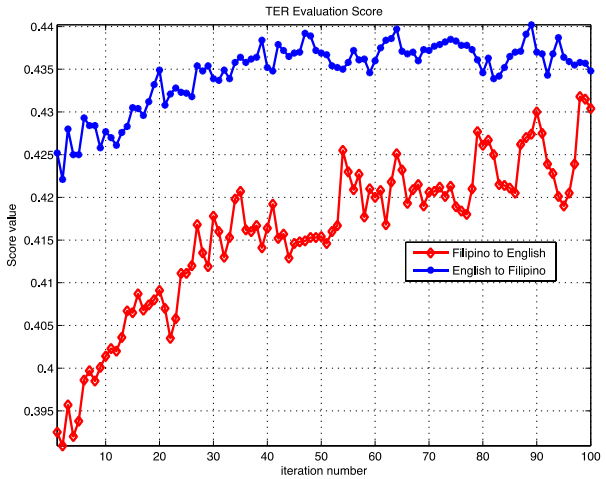


Figure 4.3: Bi-directional Filipino-to-English TER Evaluation Score using Human Feedback

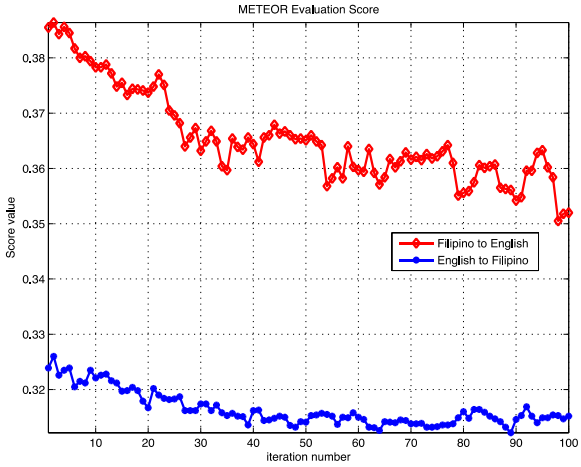


Figure 4.4: Bi-directional Filipino-to-English METEOR Evaluation Score using Human Feedback

Table 4.1: Frequency Count of Errors Based on Phrase Length using the professional translator's Target Translation

5.	**Complete**	**Under Translation**	**Over Translation**	**Equal Translation**
Eng-to-Fil Baseline	11	26	40	23
Eng-to-Fil 101st APE	20	26	30	24
Fil-to-Eng Baseline	41	26	20	13
Fil-to-Eng 101st APE	28	24	27	21

Table 4.2: Frequency Count of the Types of Error for the Target Translation

6.	**Mistranslation**	**Parts of Speech**	**Word Order**	**Untranslated**
Eng-to-Fil Baseline	27	19	13	11
Eng-to-Fil 101st APE	30	24	14	11
Fil-to-Eng Baseline	17	14	13	15
Fil-to-Eng 101st APE	29	15	22	9

In performing error analysis and evaluation of FEBSMT, a professional translator provided 100 sentences as feedback for comparison purposes. The professional translator also provided a set of categories for classifying the errors, namely *complete*, *under translation*, *over translation*, and *equal*. For an error to fall under this set of categories, the word count of the translation output is compared against the word count of the professional translator's feedback.

A translation is *complete* if the word count for both sentences is equivalent, with the context of the sentences being the same. If the contexts of the compared sentences are different, the translation will be classified under the *equal* category. *Under translation* means that the word count of the target translation is less than the feedback, while *over translation* means it has exceeded. When a translation is classified as *under*, *over* or *equal*, the type of errors that caused the failure in translation is checked.
The four main types of translation errors that were considered are *mistranslation*, *parts of speech*, *word order*, and *untranslated*,.explained as follows:

- *Mistranslation* – This is the failure to translate a part of the source sentence to its correct translation, or when the target sentence is unintelligible.
- *Parts of Speech* – This error occurs when there is a wrong usage of pronouns, tenses or verb agreements.
- *Word Order* – This error occurs when a word is misplaced in a sentence.
- *Untranslated* – This error occurs when words from the source sentence are retained in the target translation.

As shown in Table 4.1, the results for both the baseline and the 101st incremental training of English-to-Filipino are close to each other. Although, the 101st incremental training managed to obtain more complete sentences, it lessened the over translated sentences, and increased the number of equally translated sentences by 1. However, the number of mistranslation, parts of speech, and word order slightly increased based on the results shown in Table 4.2. This can denote that while the errors increased for some sentences, there were also some sentences, which were completely fixed by the 101st incremental training. On the contrary, the result for the Filipino-to-English translation showed an obvious deterioration as the number of complete sentences decreased greatly while the number of mistranslation and word order errors increased. Also evident in Table 4.4, the most frequent occurring errors are mistranslation and parts of speech errors.

6.1 Quantity-Based Human Feedback

Another experiment was conducted in order to observe the effect of merging several feedback data of APE and treating them as a single feedback data. There were a total of 15 APE incremental training systems combined together consisting of the 1st to the 15th APE in a single APE incremental training. This is for better comparison against the 15th APE, which is the highest scoring APE for the English-to-Filipino. A single APE contains 1,000 sentences as their training data; hence, there are a total of 15,000 feedback sentences trained for English-to-Filipino in this experiment. On the contrary, there were 6 incremental trainings of APE combined together for Filipino to English, which consists of 6,000 sentences in total.

Table 4.3: English-to-Filipino Comparison of 15th and Merged APE

7.	BLEU	NIST	TER	METEOR
15th APE	0.3333	6.5343	0.4293	0.3218
Merged (1st to 15th APE)	0.2726	6.0478	0.4750	0.3036

Table 4.4: Filipino-to-English Comparison of 6th and Merged APE

8.	BLEU	NIST	TER	METEOR
6th APE	0.3795	7.4114	0.3995	0.3795
Merged (1st to 6th APE)	0.3610	7.2671	0.4100	0.3765

As a result, training the feedback data in smaller sets is still better than training them in larger sized data. The scores in Table 4.3 and Table 4.4 show that the 6th and 15th APE got higher precision scores for the four metrics compared to their merged counterpart. This denotes that doing incremental training allowed more word matches, clustered words, and lesser corrections needed to be done to match the reference sentences. With lesser amount of data, the system is able to learn better as the weighted sum of the probability values in the language model is taken. However, given that the feedback data is trained as a whole, the system will only take in the current probability value causing a poor translation quality. Applying interpolation and combination methods limits the probability increase or decrease of n-grams in the language model and preserves previous phrase pairs, which limits the amount of changes done to the translation.

8.1.1 Unique Quantity-Based Human Feedback

Table 4.5: English-to-Filipino Comparison of 8th and Merged APE

	BLEU	NIST	TER	METEOR
8th APE	0.3300	6.4999	0.4305	0.3195
Merged (1st to 8th APE)	0.3352	6.5512	0.4271	0.3227

Table 4.6: Filipino-to-English Comparison of 8th and Merged APE

	BLEU	NIST	TER	METEOR
8th APE	0.3743	7.3323	0.4049	0.3753
Merged (1st to 8th APE)	0.3892	7.5288	0.3903	0.3887

The previous 100 APE incremental training phases were trained between baseline development data that was first translated in the baseline and corresponding human post-edited feedback. However, the baseline development data contains duplicates that could result to repetitions in the training data of the APE phases. Since having more repetitions increases the probability values for both the language model and the phrase table, it is necessary to observe how unique sets of feedback will improve the translation when trained in the APE setting.

Of the total of 4,406 sentences in the gathered human feedback for both English and Filipino, there were a total of 4,111 unique English sentences and 4,174 unique Filipino sentences. In order to also investigate the effect of changing the size of human feedback for each incremental training iteration, the size of the training data for each iteration was changed from 1000 to 500 sentences. Eight sets of APE incremental training data were built. The goal of the experiment is to compare between the 8 APE incremental training phases and the merged APE, composed of the same 8 phases of the APE. In all other aspects, this experiment was similar to that of the Quantity-Based Human Feedback (Section 4.3). The merged APE in the previous experiment contained duplicates, which was a possible factor for the lower translation quality because duplicate entries increase the probabilities of wrong translation pairs. With these ~4K unique sentences, the merged APE can be analyzed without the factor of incorrect duplicate translation pairs.

In Table 4.5 and Table 4.6, for both English-to-Filipino and Filipino-to-English translations, the merged APE incremental training phase has better evaluation metric scores compared to the 8 separate APE incremental training phases. The main reason for the increase in score is its uniqueness. Since there were no duplicates, the APE phase was able to learn all sentences equally wherein it calculated a more accurate computation of the probabilities. The number of training data for an APE phase does not directly mean the decrease in translation quality.

Table 4.7: English-to-Filipino Unique Incremental APE Phases

9.	BLEU	NIST	TER	METEOR
1st APE	0.3316	6.5162	0.4317	0.3194
2nd APE	0.3282	6.4722	0.4347	0.3188
3rd APE	0.3239	6.4550	0.4350	0.3176
4th APE	0.3323	6.5203	0.4308	0.3215
5th APE	0.3313	6.152	0.4304	0.3212
6th APE	0.3323	6.5214	0.4301	0.3216
7th APE	0.3345	6.5402	0.4287	0.3220
8th APE	0.3352	6.5512	0.4271	0.3227

Table 4.8: Filipino-to-English Unique Incremental APE Phases

10.	BLEU	NIST	TER	METEOR
1st APE	0.3714	7.3506	0.4028	0.3798
2nd APE	0.3729	7.3610	0.4036	0.3785
3rd APE	0.3783	7.4151	0.3991	0.3821
4th APE	0.3816	7.4386	0.3971	0.3833
5th APE	0.3796	7.4249	0.3976	0.3827
6th APE	0.3804	7.4254	0.3968	0.3832
7th APE	0.3834	7.4539	0.3950	0.3850
8th APE	0.3892	7.5288	0.3903	0.3887

In addition, for a single unique APE incremental training phase, more unique sentences would entail better machine translation quality.

The automated evaluation scores are listed in Table 4.7 and Table 4.8. The general trends for BLEU and NIST for both English-to-Filipino and Filipino-to-English translations are shown in Figure 4.5 and Figure 4.6.

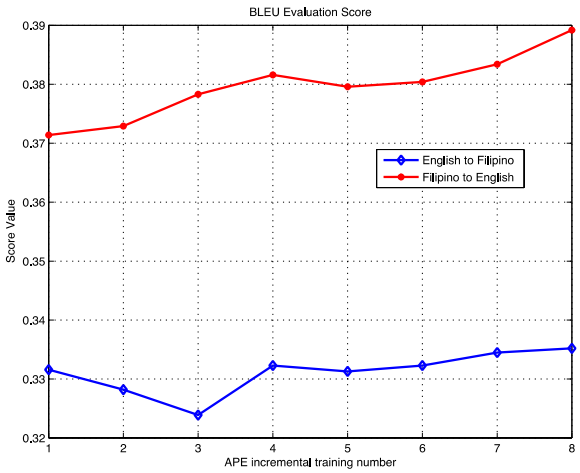


Figure 4.5: Comparison of BLEU Evaluation Score of Unique Incremental APE Phases

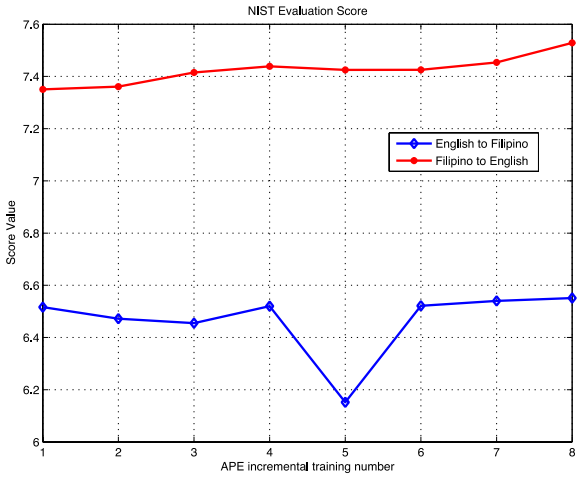


Figure 4.6: Comparison of NIST Evaluation Score of Unique Incremental APE Phases

The performance trends of the APE systems trained with unique feedback data, as shown in Figure 4.5 and Figure 4.6, are steadily increasing and are more stable. Although it appears that training with unique feedback data is better, the performance difference may be due to the way the test sentences were selected, which is purely random and did not consider the biases towards more frequently translated sentences or phrases captured using crowdsourced feedback, and on which the APE systems were incrementally trained.

11. CONCLUSION AND FUTURE WORK

In the implementation of FEBSMT, a feedback system was added to a statistical machine translation system to make updates more dynamic and responsive to quality human feedback. The four evaluation metrics namely, BLEU, NIST, METEOR, and TER were used. The system was implemented bi-directionally and both were iteratively run until convergence rates of translation scores are observed. The machine translation quality of the APEs at the onset is higher than their respective baseline evaluation scores. However, the evaluation scores soon reached its peak before decreasing gradually. This suggests that the feedback significantly affected the probability scores of the Language Model and the phrase tables, and thus affected the translations of the baseline system that are correct to begin with. It would, however, be interesting to empirically investigate the corresponding trends if the training and feedback data be made much larger by letting the system run in the long term. Furthermore, we observed that the Filipino- to-English translation has a higher machine translation quality overall, compared to the English-to-Filipino translation.

For the post-editing, the source of feedback may use the concept of crowdsourcing, wherein FEBSMT will be made available online for humans to use and provide feedback. There will be more users and the translation system will be tested thoroughly. Deploying the translation system will bring more sources of feedback and a better opportunity for the system to improve its translation. There can also be an added feature for verifying the sources of feedback and filtering out of the noisy feedback to avoid negative effects on the translation system.

12. REFERENCES

[1] Banerjee, S., Lavie, A.2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments.

[2] Doddington, G. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. Proceedings of the Human Language Technology Conference, California, 128-132.

[3] Koehn, P. 2014. MOSES. Statistical Machine Translation System, User Manual and Code Guide, Cambridge University Press.

[4] Nocon, N., Oco, N., Ilao, J., Roxas, R.2013. Philippine Component of the Network-based ASEAN Language Translation Public Service. 7th IEEE Conference Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management

[5] Papipeni, K., Roukos, S., Ward, T., and Zhu, W.J. 2002.BLEU: A Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, 311-318.

[6] Simard, M., and Foster, G.2013. PEPr: Post-Edit Propagation Using Phrase-based Statistical Machine Translation. Proceedings of the XIV Machine Translation Summit, Canada, 191-198.

[7] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.2006. A Study of Translation Edit Rate with Targeted Human Annotation.

[8] United Nations.nd. Millenium Development Goals And Beyond 2015. Goal 8: Develop A Global Partnership For Development. http://http://www.un.org/millenniumgoals/global.shtml

Parser Self-Training for Syntax-Based Machine Translation

Makoto Morishita, Koichi Akabe, Yuto Hatakoshi
Graham Neubig, Koichiro Yoshino, Satoshi Nakamrua

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma-shi, Nara, Japan
{morishita.makoto.mb1,akabe.koichi.zx8,neubig,koichiro,s-nakamura}@is.naist.jp
hatakoshi.yuto@gmail.com

Abstract

In syntax-based machine translation, it is known that the accuracy of parsing greatly affects the translation accuracy. Self-training, which uses parser output as training data, is one method to improve the parser accuracy. However, because parsing errors cause noisy data to be mixed with the training data, automatically generated parse trees do not always contribute to improving accuracy. In this paper, we propose a method for selecting self-training data by performing syntax-based machine translation using a variety of parse trees, using automatic evaluation metrics to select which translation is better, and using that translation's parse tree for parser self-training. This method allows us to automatically choose the trees that contribute to improving translation accuracy, improving the effectiveness of self-training. In experiments, we found that our self-trained parsers significantly improve a state-of-the-art syntax-based machine translation system in two language pairs.

1. Introduction

In statistical machine translation (SMT), representative methods include Phrase-Based Machine Translation (PBMT) [1], in which each phrase is translated by the translation model and reordered to the appropriate target language order, and syntax-based machine translation [2], which uses parts of syntactic parse trees for translation. While PBMT generally achieves high accuracy on language pairs with close word order such as English-French, syntax-based machine translation techniques have shown to allow for better translation accuracy on language pairs with different word order such as English-Japanese.

Among the various methods for syntax-based translation, Tree-to-String (T2S) translation [3], which uses parse trees in the source language, has been reported to achieve high translation accuracy while maintaining translation speed [4]. However, T2S translation uses parser results in the source language, so translation accuracy greatly depends on parser accuracy. One method to ameliorate this problem is Forest-to-String (F2S) translation [5], which considers multiple parse trees during the decoding process. Even F2S translation, however, is heavily affected by the accuracy of the parser used to generate the parse forest [4].

Parser *self-training* is one method to improve parser accuracy [6]. Self-training first parses unannotated sentences using an existing model, then uses these automatically generated parse trees to retrain the parser. This allows the parser to automatically adapt to the data used for self-training, increasing coverage of vocabulary or syntactic structures, and thus increasing parser accuracy. However, one downside of standard self-training methods is that automatically generated parse trees are often incorrect, which reduces their effectiveness as training data.

While there has not been much work on self-training in the context of syntax-based machine translation itself, Katz-Brown et al. [7] have proposed a method for self-training in the context of syntactic pre-ordering. In this method, which they call *targeted self-training*, they first generate multiple parse trees using a syntactic parser, then use these trees to perform pre-ordering, score the output by comparing to correct pre-ordered data, and select the parse tree that has the highest score. By using information about the correct pre-ordering to select which parse tree to use, this method has the ability to remove parse trees that result in incorrect pre-orderings, reducing noise in the training data. However, on the down side, making the manually aligned data required to apply this variety of targeted self-training is costly, limiting its applicability to situations where this data can be created.

In this paper, we propose a method for targeted self-training of parsers for syntax-based translation. The proposed method is applicable not only to pre-ordering but also syntax-based MT, and has the additional advantage that it does not require the preparation of costly hand-aligned training data because it chooses data using standard MT automatic evaluation metrics. This allows for the use of existing bilingual corpora as training data for targeted self-training, making it possible to improve parsers and F2S translation accuracy in a wider variety of fields. By carrying out experiments on targeted self-training considering machine translation accuracy, we confirmed that the proposed method signif-

icantly improves the translation accuracy of a state-of-the-art F2S system in two language pairs.

2. Tree-to-String translation

In SMT, given the source sentence $\boldsymbol{f}$, we consider the problem of finding translation $\hat{\boldsymbol{e}}$ that maximizes the posterior probability $Pr(\boldsymbol{e}|\boldsymbol{f})$

$$\hat{\boldsymbol{e}} \coloneqq \underset{\boldsymbol{e}}{\operatorname{argmax}} Pr(\boldsymbol{e}|\boldsymbol{f}). \tag{1}$$

Among the varieties of SMT, T2S translation uses source language parse tree $T_{\boldsymbol{f}}$ to disambiguate the source structure and express the hierarchical relationships between the source and target languages as rules, allowing for more accurate translation. T2S translation can be formulated as follows

$$\begin{aligned}
\hat{\boldsymbol{e}} &\coloneqq \underset{\boldsymbol{e}}{\operatorname{argmax}} Pr(\boldsymbol{e}|\boldsymbol{f}) && (2)\\
&= \underset{\boldsymbol{e}}{\operatorname{argmax}} \sum_{T_{\boldsymbol{f}}} Pr(\boldsymbol{e}|\boldsymbol{f}, T_{\boldsymbol{f}}) Pr(T_{\boldsymbol{f}}|\boldsymbol{f}) && (3)\\
&\simeq \underset{\boldsymbol{e}}{\operatorname{argmax}} \sum_{T_{\boldsymbol{f}}} Pr(\boldsymbol{e}|T_{\boldsymbol{f}}) Pr(T_{\boldsymbol{f}}|\boldsymbol{f}) && (4)\\
&\simeq \underset{\boldsymbol{e}}{\operatorname{argmax}} Pr(\boldsymbol{e}|\hat{T}_{\boldsymbol{f}}), && (5)
\end{aligned}$$

where $\hat{T}_{\boldsymbol{f}}$ is the highest probability parse tree candidate represented by the following formula:

$$\hat{T}_{\boldsymbol{f}} = \underset{T_{\boldsymbol{f}}}{\operatorname{argmax}} Pr(T_{\boldsymbol{f}}|\boldsymbol{f}). \tag{6}$$

As shown in Figure 1, translation rules used by T2S translation[1] are represented by the set of a source subtree and a target language string of words, including the replaceable variables x. In the example shown in Figure 1, x_0 and x_1 are the replaceable variables. During translation, the decoder finds the highest probability translation considering the probability of translation rules, language models, or other features. The decoder can also be used to output the n translations with the highest probability, n-best translations.

In T2S translation, by taking the source language parse tree into account, translation of long-distance word ordering can be more accurate than PBMT. However, because T2S uses the parse tree for translation, the translation accuracy greatly depends on the parser accuracy. As mentioned in the introduction, F2S translations reduce the adverse effect of parser errors by using a parse forest, which is a hyper-graph efficiently expressing a large number of parse trees. By using a parse forest for translation, the decoder can select which parse tree to use from several parse tree candidates, leading to improved translation accuracy [9]. F2S translation can be formulated as follows:

$$\langle \hat{\boldsymbol{e}}, \hat{T}_{\boldsymbol{f}} \rangle = \underset{\langle \boldsymbol{e}, T_{\boldsymbol{f}} \rangle}{\operatorname{argmax}} Pr(\boldsymbol{e}|T_{\boldsymbol{f}}) Pr(T_{\boldsymbol{f}}|\boldsymbol{f}). \tag{7}$$

[1] Specifically, T2S translation using tree transducers [8].

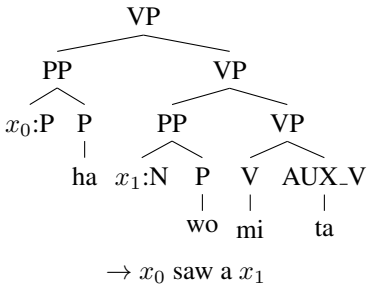


Figure 1: An example of a Japanese-to-English tree-to-string translation rule

However, even using F2S translation, the accuracy is heavily affected by the accuracy of the parser used to generate the parse forest [4]. In the following sections, we describe some methods to improve parser accuracy.

3. Parser self-training

3.1. Introduction of self-training

Parser self-training retrains the parser using the parse trees automatically generated by an existing model, allowing the parser to adapt to the data used for self-training and improve accuracy. In other words, for each sentence used for self-training, we find the highest probability parse tree $\hat{T}_{\boldsymbol{f}}$ based on Equation (6), then use this parse tree to retrain the parser.

When Charniak first proposed parser self-training, he reported that a parser using Probabilistic Context-Free Grammar (PCFG) models trained using the WSJ corpus [10] achieved no gain through self-training [11]. On the other hand, the PCFG with Latent Annotations (PCFG-LA) model, which achieves improved parsing accuracy by using latent annotations, has been reported to be improved significantly by self-training [12]. This is because the PCFG-LA has relatively high accuracy, so the automatically generated parse trees used for self-training are more accurate, and because the PCFG-LA model has more parameters than standard PCFGs, making it more likely to benefit from the increased amount of data. Based on these studies, we consider parser self-training using the PCFG-LA model.

3.2. Self-training of the parser in machine translation

As mentioned in the introduction, there is one previous study on improving translation accuracy by doing parser self-training. Katz-Brown et al. propose a method for *targeted self-training*, where trees are selected based on an extrinsic evaluation measure, and report that it is possible improve translation accuracy itself [7]. Specifically, they automatically generate several candidate parse trees, then select the candidate for which the pre-ordering result is most similar to hand-aligned correct data. This tree is then used to retrain the parser.

Formally, we define the pre-ordering function reord($T_{\boldsymbol{f}}$),

which generates a pre-ordered source language sentence $\boldsymbol{f}'$ based on a parse tree $T_{\boldsymbol{f}}$, and a score function score($\boldsymbol{f}'^{*}$, $\boldsymbol{f}'$) [13], which compares $\boldsymbol{f}'$to the reference preordered sentence $\boldsymbol{f}'^{*}$. Parse tree $\bar{T}_{\boldsymbol{f}}$, which is used in self-training, is selected from the candidate parse trees $\boldsymbol{T}_{\boldsymbol{f}}$ by the following formula

$$\bar{T}_{\boldsymbol{f}} = \underset{T_{\boldsymbol{f}} \in \boldsymbol{T}_{\boldsymbol{f}}}{\operatorname{argmax}} \operatorname{score}(\boldsymbol{f}'^{*}, \operatorname{reord}(T_{\boldsymbol{f}})) \tag{8}$$

In this paper, based on these previous studies, we propose a method for parser targeted self-training for syntax-based translation. In the following sections, we explain the detail of this method and verify its effectiveness.

4. Parser self-training for syntax-based MT

An important point that determines the effectiveness of self-training is how to select the data used to retrain the parser. In the following sections, we propose several methods to select parse trees and sentences that are effective to improve the accuracy of F2S translation.

4.1. Selecting parse trees

As described in Section 3.2, the targeted self-training proposed by Katz-Brown et al. [7], selects the most accurate parse tree from n-best candidates by comparing hand-aligned correct preordering data and automatically generated parse trees. However, constructing hand-aligned data is costly, and thus it is impractical to create large data sets for this method. To solve this problem, we propose two methods for targeted self-training using only a parallel corpus. One is to use the parse tree used in the 1-best translation selected by the decoder, and the other is to use the parse tree used in the oracle translation, which is the most similar translation to the reference translation from the n-best list selected by automatic evaluation metrics.

4.1.1. Decoder 1-best

As described in Section 2, in F2S translation, the decoder selects the parse tree used to generate the translation with the highest probability from the parse forest. A previous study has noted that the F2S decoder has the ability to select more accurate parse trees, as it uses other feature functions such as rule probabilities and language model probabilities, which cannot be considered by the baseline parser [9]. Thus, the parse tree used in the one-best translation could be more effective for self-training than the parser 1-best tree. In this case, the parse tree used in self-training is $\hat{T}_{\boldsymbol{f}}$ in Formula (7).

4.1.2. Automatic evaluation 1-best

During translation, the decoder outputs the translation that has the highest translation probability from a multitude of translation candidates. However, there are also cases in which other candidate translations, for example ones in the n-best list, are more similar to the reference translation, which indicates that they may be more accurate than the decoder 1-best translation.

We define the oracle translation $\bar{\boldsymbol{e}}$, which is the closest to reference translation $\boldsymbol{e}^{*}$ among the n-best translation candidates $\boldsymbol{E}$. In this method, we perform self-training using the parse tree that is used in the oracle translation $\bar{\boldsymbol{e}}$. By using the score function $\operatorname{score}(\cdot)$, which represents the similarity between the hypothesis and reference translation, $\bar{\boldsymbol{e}}$ is formulated as follows:

$$\bar{\boldsymbol{e}} = \underset{\boldsymbol{e} \in \boldsymbol{E}}{\operatorname{argmax}} \operatorname{score}(\boldsymbol{e}^{*}, \boldsymbol{e}). \tag{9}$$

4.2. Selecting sentences

In Section 4.1, we described methods to select, from an n-best list for a single sentence, parse trees that may be useful in self-training. However, in many cases, the correct parse tree may not be included in the n-best translation candidates, and there is a possibility of these sentences adding noise to the training data. Therefore, we further propose two methods for selecting which sentences should be used in self-training from the entirety of the training data, potentially removing sentences for which no good n-best candidate exists. One is to use sentences for which the translated sentence's automatic evaluation score exceeds a threshold, and the other is to use sentences that have a large score increase between the decoder 1-best and oracle translation.

4.2.1. Automatic evaluation threshold

There are some sentences in the corpus that are not translated accurately by the MT system, and the score of automatic evaluation metrics decreases. The cause of low evaluation scores could be for the the following reasons:

- An incorrect parse tree has been used in the translation.
- The translation model does not sufficiently cover the source sentence's vocabulary or phrases.
- The reference translation, which is used to calculate the automatic evaluation score, is a free or incorrect translation, and the system cannot output a similar translation.

In such cases, the score of automatic evaluation metrics will be low even for the oracle translation. Because the F2S decoder cannot select the correct trees or the evaluation scores are not reliable, these data are more likely to have noisy oracle trees for training, and thus it is potentially beneficial to exclude these trees from the training data. For these reasons, we propose a sentence selection method that uses only sentences that achieve scores over a threshold, which can be expected to have more accurate parse trees. The set of sentences for self-training is defined as follows, where t is the threshold, $\boldsymbol{e}^{*(i)}$ is the reference translation of the sentence i, $\bar{\boldsymbol{e}}^{(i)}$ is the oracle translation of the sentence i, $\bar{\boldsymbol{E}}$ is

the set of all oracle translations, and score(e) is the automatic evaluation score function

$$\{i \mid \text{score}(\boldsymbol{e}^{*(i)}, \bar{\boldsymbol{e}}^{(i)}) \geq t,\ \bar{\boldsymbol{e}}^{(i)} \in \bar{\boldsymbol{E}}\}. \quad (10)$$

4.2.2. Automatic evaluation gain

Next, we focused on the difference between the automatic evaluation score of the decoder 1-best and oracle translation. In the case that the parse forest output by the parser has incorrect probabilities for the parse trees, in many cases, the decoder will select the incorrect parse tree and output the wrong translation. On the other hand, the oracle translation is more likely to have used a correct parse tree from the parse forest. Therefore, by using the parse trees used in oracle translations as training data in these cases, it may be possible to improve the parser's probability estimates. This will result in the system using the self-trained parser tending to output the correct translation as a 1-best, improving translation accuracy.

To select the sentences, we define the function gain$(\bar{\boldsymbol{e}}^{(i)}, \hat{\boldsymbol{e}}^{(i)})$, which represents the gain between the score of 1-best translation $\hat{\boldsymbol{e}}^{(i)}$ and oracle translation $\bar{\boldsymbol{e}}^{(i)}$, then selects the sentences with highest gain as in Formula (10). The function gain$(\bar{\boldsymbol{e}}^{(i)}, \hat{\boldsymbol{e}}^{(i)})$ is formulated as follows:

$$\text{gain}(\bar{\boldsymbol{e}}^{(i)}, \hat{\boldsymbol{e}}^{(i)}) = \text{score}(\bar{\boldsymbol{e}}^{(i)}) - \text{score}(\hat{\boldsymbol{e}}^{(i)}). \quad (11)$$

In addition, in this method, in order to ensure that the sentence length distribution of the training data is similar to that of the entire corpus, we use the following formula proposed by Gascó et al. [14], to ensure that the length distribution of the selected sentences is similar to that of the overall corpus distribution.[2] In this formula, $|\boldsymbol{e}|$ is the length of target language sentence $\boldsymbol{e}$, $|\boldsymbol{f}|$ is the length of source language sentence $\boldsymbol{f}$, $N_c(|\boldsymbol{e}| + |\boldsymbol{f}|)$ is the number of sentences in the corpus with length $|\boldsymbol{e}| + |\boldsymbol{f}|$, and N_c is the number of sentences in the entire corpus

$$p(|\boldsymbol{e}| + |\boldsymbol{f}|) = \frac{N_c(|\boldsymbol{e}| + |\boldsymbol{f}|)}{N_c}. \quad (12)$$

The number of sentences selected for the self-training set is formulated as follows, where $N_t(|\boldsymbol{e}| + |\boldsymbol{f}|)$ is the number of sentences in the self-training set with length $|\boldsymbol{e}| + |\boldsymbol{f}|$, and N_t is the number of sentences in the entire self-training set

$$N_t(|\boldsymbol{e}| + |\boldsymbol{f}|) = p(|\boldsymbol{e}| + |\boldsymbol{f}|)N_t. \quad (13)$$

5. Experiments

5.1. Experimental setup

In the experiments, we focused on Japanese-English and Japanese-Chinese translation. Because the amount of hand-labeled Japanese parse tree data is less than that available for English, the Japanese parser is prone to parse errors. As the translation data, we use ASPEC,[3] which is a parallel corpus of scientific papers abstracts. The number of sentences in ASPEC is shown in Table 1.[4] As a state-of-the-art baseline for verifying the effect of self-training, we use the system developed by Neubig [15],[5] which was the most accurate system on the Workshop on Asian Translation (WAT) 2014 [16]. We use Travatar [17] as a Forest-to-String decoder. As a parser, we use the PCFG-LA parser Egret,[6] and train a baseline model on a phrase-structure version of the Japanese Dependency Corpus (JDC) [18], which has about 7000 sentences. Forests were pruned to remove hyper-edges which do not appear in the 100 n-best trees. Egret sometimes fails to output a parse tree, and in this case, we remove the failed sentences. We evaluate the accuracy by using two automatic evaluation metrics, BLEU [19] and RIBES [20], and to evaluate the accuracy for each sentence in sentence or oracle selection, we use BLEU+1 [21]. For self-training data, we add the data selected from the ASPEC training data to the JDC trees. The training data for the translation systems is parsed using the standard JDC model, and the self-trained models are used only to parse the development and test corpora at test time.[7] We verify statistical significance using the bootstrap resampling method [22]. In the next section, we compare the following parser self-training methods:

Table 1: The number of sentences in ASPEC

	Train	Dev	DevTest	Test
Ja-En	2,000,000	1,790	1,784	1,812
Ja-Zh	672,315	2,090	2,148	2,107

Parser 1-best
: As in Formula (6), we use the 1-best parse trees for self-training. We select the sentences randomly from the corpus.[8]

MT 1-best
: As described in Section 4.1.1, we input the parse forest to the decoder, and use the parse trees used in the 1-best translation. We select the sentences randomly from the corpus as in Parser 1-best.

Oracle
: As described in Section 4.1.2, we input the parse forest to the decoder, output unique 500-best hypotheses and use the parse tree corresponding to the translation that has the highest BLEU+1 score in this n-best list. We

[2] The BLEU gain approach was not effective if we did not use this technique, as it tends to select only short sentences where small changes in wording cause large changes in evaluation scores.

[3] http://lotus.kuee.kyoto-u.ac.jp/ASPEC

[4] ASPEC actually has 3.0 million Ja-En training sentences, but because the data was automatically aligned, we use only the highest-confidence 2.0 million sentences to maintain the quality of the training data.

[5] http://github.com/neubig/wat2014

[6] http://code.google.com/p/egret-parser

[7] It may be possible to further improve translation accuracy by re-parsing the training data, but this comes at a significant computational cost, so in this work we only experiment with re-parsing the development and test corpora.

[8] While a large corpus is available, training the parser using the entire corpus is computationally expensive, so we randomly subsample a training corpus.

Table 2: Experiment results of Japanese-English translation

	Sentence selection	Tree selection	Ja-En			Ja-Zh		
			Sent	BLEU	RIBES	Sent	BLEU	RIBES
(a)	—	—	—	23.83	72.27	—	29.60	81.32
(b)	Random	Parser 1-best	96k	23.66	71.77	129k	29.75	‡ 81.55
(c)	Random	MT 1-best	97k	23.81	72.04	130k	29.76	‡ 81.53
(d)	Random	BLEU+1 1-best	97k	23.93	72.09	130k	‡ 29.89	‡ 81.66
(e)	BLEU+1 $\geq$ 0.7	BLEU+1 1-best	206k	‡ 24.27	72.38	240k	‡ 29.86	‡ 81.60
(f)	BLEU+1 $\geq$ 0.8	BLEU+1 1-best	120k	‡ 24.26	72.38	150k	‡ 29.91	81.47
(g)	BLEU+1 $\geq$ 0.9	BLEU+1 1-best	58k	‡ 24.26	72.49	82k	† 29.86	‡ 81.60
(h)	BLEU+1 Gain	BLEU+1 1-best	100k	† 24.22	72.32	100k	† 29.85	‡ 81.59
(i)	BLEU+1 $\geq$ 0.8 (Ja-En)	BLEU+1 1-best	—	—	—	120k	† 29.87	† 81.58

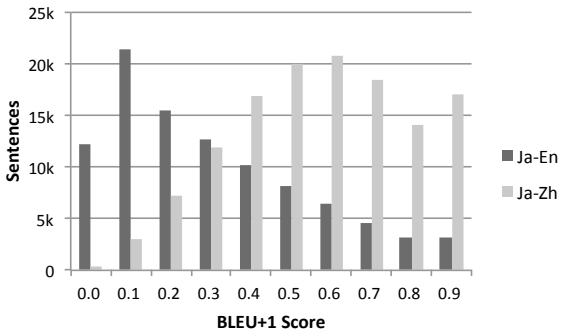


Figure 2: BLEU+1 score distribution of translations in Table 2 (d).

select the sentences randomly from the corpus as in Parser 1-best.

Oracle (BLEU+1≥t)
As described in Section 4.2.1, among the oracle translations and parse trees, we only use the sentences for which the BLEU+1 score exceeds the threshold t.

BLEU+1 Gain
As described in Section 4.2.2, among the oracle translations and parse trees, we use only the sentences which have a large difference of BLEU+1 score between 1-best and oracle translations. In this method, we maintain the sentence length distribution by using Formula (12) and (13).

It should be noted that when selecting the sentences randomly, we select $1/20$ of all training data in Japanese-English, $1/10$ in Japanese-Chinese translation. In BLEU+1 Gain, we select the top 100k sentences, which is a similar number of sentences as used in the other methods.

5.2. Experiment results

Table 2 shows the experimental results for Japanese-English and Japanese-Chinese translation. The dagger symbol in the table indicates that the translation accuracy of the proposed method is significantly higher than the baseline ($\dagger : p < 0.05$, $\ddagger : p < 0.01$). In Table 2 (b), (c), (d), the sentences for self-training are the same except where Egret fails to parse.[9] In Table 2, "Sent" indicates the number of sentences added through self-training and does not include the existing JDC data. In our analysis, we mainly focus on the BLEU score results, because we used BLEU+1 as a criterion when picking sentences for self training. Based on these results, we answer the following research questions:

- Is targeted self training through parse tree selection (Section 4.1) effective in improving translation results?
- Can sentence selection (Section 4.2) further reduce noise and improve accuracy?
- Is self-training language dependent, or portable across target languages?

Effect of Tree Selection: First, we can see that the method of using parser 1-best trees as self-training data did not achieve a BLEU score improvement in Ja-En and Ja-Zh translation (Table 2 (b)). Additionally, while in the MT 1-best method, the accuracy is improved compared to Parser 1-best in the Ja-En experiment, there is no improvement compared to the baseline system. In the Ja-Zh experiment, MT 1-best is almost the same as parser 1-best (Table 2 (c)). We manually analyzed the parse trees that have been used for self-training in these methods, and while there are some correct trees there are also many incorrect trees, which likely disturbed the training.

Next looking at the BLEU+1 1-best scores (Table 2 (d)), we can see that by selecting parse trees that were used in the oracle translations, BLEU scores slightly improved in both Ja-En and Ja-Zh experiments, with the Ja-Zh system significantly outperforming the baseline. Figure 2 shows the BLEU+1 score distribution of oracle translations used in self-training in this case. The label on the horizontal axis repre-

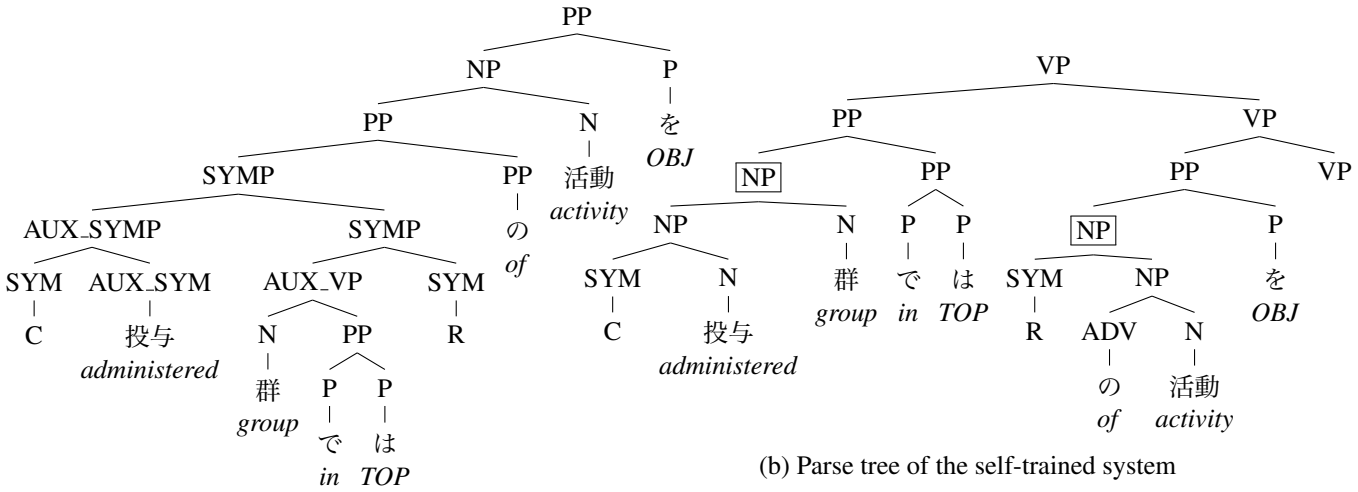
[9] In the methods except (b), we use Ckylark [23] trained by JDC as an alternative parser when Egret fails to parse.

Table 3: Self-trained Japanese parser accuracy

	Sentence selection	Tree selection	Recall	Precision	F-Measure
(a)	—	—	84.88	84.77	84.83
(b)	Random	Parser 1-best	86.52	86.41	† 86.46
(c)	BLEU+1 $\geq$ 0.8	BLEU+1 1-best	88.13	88.01	‡ 88.07

Table 4: An example of an improvement in Japanese-English translation

Source	C 投与 群 で は R の 活動 を ２４０ 分 に わた って 明らか に 増強 した 。
Reference	in the C - administered group , thermal reaction clearly increased the activity of R for 240 minutes .
Baseline	for 240 minutes clearly enhanced the activity of C administration group R .
BLEU+1≥0.8	for 240 minutes clearly enhanced the activity of R in the C - administration group .



(a) Parse trees of the baseline system

(b) Parse tree of the self-trained system

Figure 3: An example of an improvement in parsing result

sents the number of sentences which have BLEU+1 scores greater than x and less than $x + 0.1$, where x is the label. As can be seen from the figure, there are many sentences where even the oracle has a low score, motivating the sentence selection methods presented in Section 4.2.

Effect of Sentence Selection: Next, we examine the effect of selecting sentences using the BLEU+1 score threshold. From the results, we find it effective, with translation accuracy improving especially in Ja-En experiments (Table 2 (e),(f),(g)). In the Ja-Zh experiments, the BLEU+1 score distribution of oracle translations tends to be higher than in the Ja-En translations, explaining why this method is more effective in Ja-En experiments. From this result, we can say that when doing parser self-training, it is important to remove the low accuracy parse trees and keep only high accuracy parse trees from the training data. This is particularly true when there are a large number of oracle translations with low accuracies.

Moreover, by using the data that have a large BLEU+1 score improvement between MT 1-best and oracle translations, we achieved an effect similar to that of the BLEU+1 threshold method (Table 2 (h)).

Target Language Portability: Finally, we examine what happens when a parser used for translation in one language pair, Japanese-English, is used to parse the sentences for translation in another language pair, Japanese-Chinese (Table 2 (i)). Interestingly, the improvement in this case is quite similar to the parser trained directly on the Japanese-Chinese data. Thus, the model's dependence on target language is not strong, and it may be possible to do more effective self-training by using several target languages as training data.

5.3. Example of improved translation

Table 4 shows an example of an improvement caused by self-training in Japanese-English translation. In addition, Figure

3 shows the parse trees used in the translations in Table 4. In this sentence "C 投与 群" (C administration group) and "R の 活動" (activity of R) are noun phrases. The baseline parser cannot identify noun phrases correctly, and translation is affected by this parse error. On the other hand, the self-trained parser can identify noun phrases correctly, resulting in these phrases being correctly translated.

5.4. Self-trained parser accuracy

We also performed experiments to examine the parser accuracy itself. We manually created 100 reference parse trees from the Ja-En ASPEC test data, and checked the accuracy of the baseline and self-trained parsers with respect to these trees by using Evalb.[10] Table 3 shows the experimental results. The dagger symbol in the table indicates that the F-Measure of the proposed method is significantly higher than the baseline ($\dagger : p < 0.05$, $\ddagger : p < 0.01$).

Here, we can see that the parser 1-best method achieved significantly higher accuracy than the baseline at the 95% level. In addition, our proposed targeted self-trained parser could achieve a further significant gain in accuracy. These results show that our proposed targeted self-training methods improve not only MT results, but also parser accuracy itself.

6. Conclusion

In this study, we proposed a targeted self-training method for syntactic parsers used in syntax-based MT, and verified its effect on T2S translation. We performed experiments on Japanese-English and Japanese-Chinese translation and found that by using the self-trained parser that we were able to achieve a significant improvement in the accuracy of a state-of-the-art translation system. Moreover, we found that the model self-trained by Japanese-English sentences can also contribute to more accurate Japanese-Chinese translations.

Our future work includes verifying that this method can be used for other languages pairs. Moreover, the experimental results suggest that the effect of self-training does not heavily depend on the source language, and thus it may be possible to improve the translation accuracy by applying self-training over data from multiple languages pairs. Furthermore, we will test the effect on translation accuracy when performing multiple iterations of parser self-training, or using the self-trained parser to re-parse the training data.

7. Acknowledgments

Part of this research was supported by JSPS KAKENHI Grant Number 25730136 and 24240032.

[10] http://nlp.cs.nyu.edu/evalb

8. References

[1] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. HLT*, 2003, pp. 48–54.

[2] K. Yamada and K. Knight, "A syntax-based statistical translation model," in *Proc. ACL*, 2001, pp. 523–530.

[3] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation," in *Proc. ACL*, 2006, pp. 609–616.

[4] G. Neubig and K. Duh, "On the elements of an accurate tree-to-string machine translation system," in *Proc. ACL*, 2014, pp. 143–149.

[5] H. Mi, L. Huang, and Q. Liu, "Forest-based translation," in *Proc. ACL*, 2008, pp. 192–199.

[6] D. McClosky, E. Charniak, and M. Johnson, "Effective self-training for parsing," in *Proc. HLT*, 2006, pp. 152–159.

[7] J. Katz-Brown, S. Petrov, R. McDonald, F. Och, D. Talbot, H. Ichikawa, M. Seno, and H. Kazawa, "Training a parser for machine translation reordering," in *Proc. EMNLP*, 2011, pp. 183–192.

[8] J. Graehl and K. Knight, "Training tree transducers," in *Proc. HLT*, 2004, pp. 105–112.

[9] H. Zhang and D. Chiang, "An exploration of forest-to-string translation: Does translation help or hurt parsing?" in *Proc. ACL*, 2012, pp. 317–321.

[10] 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.

[11] E. Charniak, "Statistical parsing with a context-free grammar and word statistics," in *Proc. AAAI*, 1997, pp. 598–603.

[12] Z. Huang and M. Harper, "Self-training PCFG grammars with latent annotations across languages," in *Proc. EMNLP*, 2009, pp. 832–841.

[13] D. Talbot, H. Kazawa, H. Ichikawa, J. Katz-Brown, M. Seno, and F. Och, "A lightweight evaluation framework for machine translation reordering," in *Proc. WMT*, 2011, pp. 12–21.

[14] G. Gascó, M.-A. Rocha, G. Sanchis-Trilles, J. Andrés-Ferrer, and F. Casacuberta, "Does more data always yield better translations?" in *Proc. ACL*, 2012, pp. 152–161.

[15] G. Neubig, "Forest-to-string SMT for asian language translation: NAIST at WAT2014," in *Proc. WAT*, 2014.

[16] T. Nakazawa, H. Mino, I. Goto, S. Kurohashi, and E. Sumita, "Overview of the 1st Workshop on Asian Translation," in *Proc. WAT*, 2014.

[17] G. Neubig, "Travatar: A forest-to-string machine translation engine based on tree transducers," in *Proc. ACL Demo Track*, 2013, pp. 91–96.

[18] S. Mori, H. Ogura, and T. Sasada, "A Japanese word dependency corpus," in *Proc. LREC*, 2014, pp. 753–758.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. ACL*, 2002, pp. 311–318.

[20] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada, "Automatic evaluation of translation quality for distant language pairs," in *Proc. EMNLP*, 2010, pp. 944–952.

[21] C.-Y. Lin and F. J. Och, "Orange: a method for evaluating automatic evaluation metrics for machine translation," in *Proc. COLING*, 2004, pp. 501–507.

[22] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proc. EMNLP*, 2004, pp. 388–395.

[23] Y. Oda, G. Neubig, S. Sakti, T. Toda, and S. Nakamura, "Ckylark: A more robust PCFG-LA parser," in *Proc. NAACL*, 2015, pp. 41–45.

Risk-aware Distribution of SMT Outputs for Translation of Documents Targeting Many Anonymous Readers

Yo Ehara[†], *Masao Utiyama*[‡], *Eiichiro Sumita*[‡]

†: Tokyo Metropolitan University, Tokyo, Japan
‡: National Institute of Information and Communications Technology, Kyoto, Japan
†:ehara@tmu.ac.jp, ‡:{mutiyama,eiichiro.sumita}@nict.go.jp

Abstract

Web documents and news articles are typically written for many anonymous readers. Thus, when translating such documents, the total quality of translations distributed to the entire readers should be considered. Previous statistical machine translation studies have focused on selecting the best translation from N-best candidates. However, when dealing with many readers, it is not necessary to identify the best translation. Our key idea is to distribute all good candidate translations to the readers and improve the total quality of the translations. We simulated a case with $1,000$ news document readers and showed statistically significant gain in sentence-level BLEU scores averaged over those readers.

1. Introduction

Web documents and news articles are typically written for many anonymous readers. Unlike documents that target specific readers such as mails and letters, the number of readers of web documents and news articles cannot be determined in advance. When translating documents that target a large number of readers, our goal is to improve the total quality of all translated documents rather than improving the translation quality of a single document.

Previous statistical machine translation (SMT) studies have focused on selecting one best translation from many candidate translations and have not considered the number of readers [1, 2]. Selecting one translation frees us from considering the number of readers because a target language reader usually only reads one translation of source language material. Thus, selecting a single translation is an effective strategy if a good translation is always selected as the best translation.

However, current SMT techniques cannot always identify the *actual* best translation from candidate translations. In many cases, even when there is a good translation among the candidates, SMT systems frequently rank bad translations higher than good translations. In other words, the strategy that attempts to find a single best translation risk selecting poor translations, even when good candidate translations are available. Thus, it is preferable to select multiple candidate translations when the task setting allows us to do so.

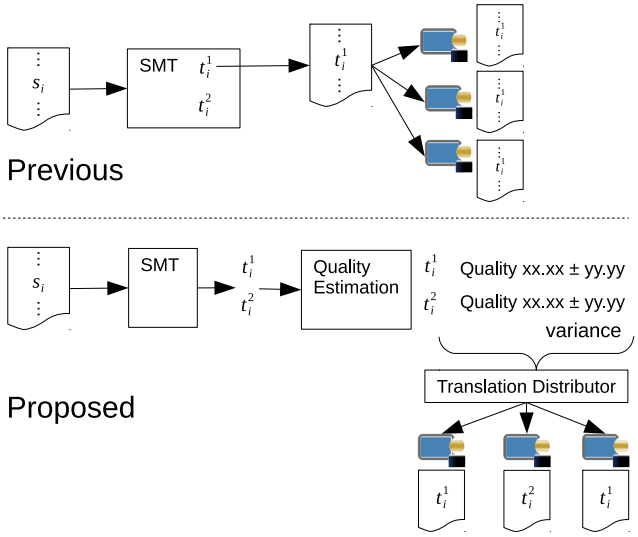


Figure 1: *Schematic Comparison of Previous and Proposed Approach*

We propose an approach for distributing multiple translation candidates when translating documents for many anonymous readers, such as web documents and news articles. Our key idea is to distribute all seemingly good candidate translations. A schematic diagram of the proposed approach is presented in Figure 1. In a previous approach [1], for source sentence s_i, an SMT system produces and ranks several candidate translations of s_i, i.e., t_i^1 and t_i^2. Only the top ranked translation, t_i^1, is used; therefore, the three readers only read t_i^1. However, it is possible that the actual quality of t_i^1 is lower than that of t_i^2. In this situation, the readers do not have access to the best translation. In the proposed approach, we perform *quality estimation* (QE) for the quality and quality variance of each candidate translation. Considering both quality and variance, we calculate *rates* that determine how many of the entire readers should read each candidate translation. Then, using these rates, we *distribute* candidate translations to all the readers. As can be seen in Figure 1, the

[†]This work was conducted when the first author was employed at NICT. We would like to thank anonymous reviewers for their insightful comments that helped us to improve this paper. This paper was proofread under the funding of NICT and JSPS KAKENHI Grant Number 15K16059.

Table 1: Motivating Example using Japanese to English Translation; "inu" means dog or dogs, and "naku" has multiple meanings. BLEU[3] is a widely used translation quality metric.

Legend	Context	BLEU
Source	inu/N ga/SUBJ-marker naku/V .	-
1st best	A dog cries.	50.8
2nd best	A dog barks.	100.0
3rd best	Dog weeps.	38.5
Reference	A dog barks.	100.0

Table 2: BLEU scores of Translations Distributed to Each Reader (when "1st best" candidate is distributed to all four readers)

	Candidate to be Distributed	BLEU
Reader1	1st	50.8
Reader2	1st	50.8
Reader3	1st	50.8
Reader4	1st	50.8
Average	-	50.8

Table 3: BLUE scores of Translations Distributed to Each Reader (when "2nd best" candidate is distributed to one reader and "1st best" is distributed to the other three readers)

	Candidate to be Distributed	BLEU
Reader1	1st	50.8
Reader2	2nd	100.0
Reader3	1st	50.8
Reader4	1st	50.8
Average	-	63.1

proposed method distributes t_i^1 to two readers and t_i^2 to one reader. In this example, if the quality of t_i^1 is lower than that of t_i^2, the average quality of the three translations distributed to the three readers is improved.

We explain our motivation using the example in Table 1. In this example, we want to translate the Japanese sentence "inu ga naku" (A dog barks.) to English. Here "inu" translates as dog or dogs, and "ga" is a subject marker that does not need to be translated. Translating the verb "naku" is problematic because it is ambiguous in English; "naku" means to cry, to bark, and to weep.

Suppose an SMT system translates this Japanese source sentence to English and that the top three translations are

Table 4: BLUE scores of Translations Distributed to Each Reader (when "3rd best" candidate is distributed to one reader and "1st best" is distributed to the other three readers.)

	Candidate to be Distributed	BLEU
Reader1	1st	50.8
Reader2	1st	50.8
Reader3	1st	50.8
Reader4	3rd	38.5
Average	-	47.7

those shown in Table 1. Moreover, suppose there are *four* readers. If we distribute the "1st best" candidate in Table 1 to all four readers, the baseline average BLEU [3] score, a widely used metric for translation quality, is 50.8 (Table 2). Because we rely only on the "1st best" candidate, if this candidate's quality is low, the translation quality will be affected.

In contrast, considering the risk that the SMT system may fail to identify the actual best translation, we can distribute other candidates to a small number of readers. For example, as shown in Table 3, if we distribute the "2nd best" translation to one reader randomly, we can achieve an average BLEU score of 63.1, which is a great improvement compared to distributing the "1st best" candidate to all readers.

However, avoiding the risks associated with SMT systems in this manner does not always achieve good results. For example, as can be seen in Table 4, if we distribute the "3rd best" translation to one reader randomly, the average BLEU score is 47.7, which is less than the baseline average BLEU score (50.8; Table 2).

Thus, to improve performance in averaged quality, we need to 1) estimate (predict) quality of candidates accurately without a reference translation, and 2) optimize and determine the risks associated with considering both successful and unsuccessful cases.

We conducted simulation experiments to evaluate the proposed approach. In these simulation experiments, an SMT system distributes translations to $1,000$ readers. We found that the proposed approach consistently and significantly outperform the previous approach.

The contributions of this study are summarized as follows.

- We propose an approach for distributing translation candidates to readers when documents with many readers such as web documents and news articles are translated.

- Our key idea is to use all translation candidates rather than using only the top candidate by considering the possibility that the top ranked candidate is not actually the best translation.

- Our experimental results show that the proposed approach consistently outperforms the baseline approach in which only the top candidate is distributed to all readers.

The remainder of this paper is organized as follows. Section 2 differentiates our task from previous studies. Section 3 describes how to estimate quality considering risks. Section 4 explains the key idea of the proposed approach: how to use the estimated quality and its risk to distribute translations. Section 5 describes the experimental settings. Section 6 and Section 7 present quantitative and qualitative results, respectively. A discussion is presented in Section 8, and the paper is concluded in Section 9.

2. Related Work

Our approach is closely related to a quality estimation (QE) task. In this approach, the QE task estimates the quality of a given source text and its translation without a reference translation [4, 5]. From a machine learning perspective, a QE task is generally categorized as a regression problem [6]. A regression problem differs from typical classification problems, such as those that apply support vector machine (SVM) techniques, in that it tries to predict real values while the latter tries to predict classes. Many regression algorithms have been applied to QE tasks, e.g., SVM-based regression [7] and Gaussian process (GP) regression [8, 6, 9].

In addition to predicted scores, a GP can output their variances [10]; however, SVM-based regression algorithms can only output predicted scores and cannot output their variances. More precisely, SVMs can output confidence values; however, such values cannot be interpreted as variances. Although GPs can output variances, most QE systems that use a GP only use the predicted scores.

QE tasks can also be categorized by the source text unit used to estimate quality: words, sentences, or documents. This study uses sentences because they are the most widely used and studied [7]. However, the proposed approach is also applicable to words or documents. To use other types of source text units, we simply switch sentences in Figure 1 to another unit type.

Our task is also related to another previous approach, i.e., system combination [11, 12]. Given single best translations from multiple SMT systems, system combination techniques attempt to output a more sophisticated single translation by combining the given translations. Like the system combination approach, the proposed approach deals with multiple translations for a given source text.

However, the proposed task clearly differs from system combination in both objective and outputs. The objective of the proposed task is to distribute given translations to readers considering the risk in translation quality. In contrast, the goal of the system combination approach is to refine translations. In the proposed task, a translation distributed to a reader is one of the input translations. In contrast, the translation output by a system combination technique can be very different from the input translations because its objective is to refine translations.

The system combination approach and the proposed approached can be aggregated to create a new system. Given a source text, suppose a system-combination system can output *multiple* sophisticated translations rather than a single best translation. Then, the proposed approach can input the sophisticated translations and distribute them to readers. Note that, for simplicity, we do not focus on this aggregated system; however, being able to create an aggregated system implies that our task is independent of the system combination tasks.

Re-ranking candidates to find the best translation candidate has been addressed in a previous study [13]. However, unlike our goal, this study does not aim to distribute translation candidates.

3. Gaussian Process-based Quality Estimation

Here we explain how to estimate the quality of given translations considering risks in quality. As described in Section 2, we use a GP to estimate quality and its risk simultaneously because a GP can output variance in addition to quality, and this variance encodes the quality's risk.

We introduce the notations used to explain the GP. Our notations are based on a previously QE study that used a GP [6]; however, this study used a GP for multitask learning, a purpose very different from ours.

We model the proposed task as a regression problem where the training data is given as M pair $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$. Here $\mathbf{x}_i \in \mathbb{R}^d$ denotes a d-dimensional feature vector constructed from a pair of source sentences and its translation. $\mathbf{x}_i \in \mathbb{R}^d$ encodes linguistic features taken from the pair. $y_i \in \mathbb{R}$ is a response variable, which is the gold standard in regression problems. It numerically encodes the translation quality, i.e., how good the translation is for the source sentence in the i-th source sentence-translation pair. For y_i in QE, typically, a manual quality assessment such as post-editing time or a Likert score is used. However, to the best of our knowledge, no dataset with manually assessed quality for N-best output of an SMT system exists. Therefore, we have used sentence BLEU scores implemented in the Moses [2] toolkit [2].

The goal of the GP is to predict y_* for an unseen test sample $\mathbf{x}_*$ given the training data $\mathcal{D}$. The GP performs this prediction by integrating over a functional space as follows. Intuitively, this means that all possible regressor functions f within the functional space are considered in the GP.

$$p(y_*|\mathbf{x}_*, \mathcal{D}) = \int_f p(y_*|\mathbf{x}_*, f)p(f|\mathcal{D}) \quad (1)$$

In (1), function f is defined as follows.

$$f(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}')) \quad (2)$$

[2] http://www.statmt.org/moses/

(2) has two parameters. The first is the mean function $\mathbf{0}$, which simply implies that the function f is normalized to 0. The key component in (2) is k, a *covariance kernel function*, which intuitively encodes the closeness of $\mathbf{x}$ and $\mathbf{x}'$.

A typical covariance kernel function is a radial basis function (RBF), which is expressed as follows[3].

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^\top A^{-1}(\mathbf{x}-\mathbf{x}')\right) \quad (3)$$

There are two hyperparameters in (3), σ_f and A. σ_f is a scalar that determines the overall size of the variances. $A = \mathrm{diag}(\mathbf{a})$ is a diagonal matrix that determines the weight of each feature; the importance of the i-th feature increases as a_i increases. Typically, $\mathbf{a}$ is defined as $\mathbf{a} = \sigma_\ell^2\mathbf{1}$ where $\mathbf{1}$ is a vector of appropriate size whose elements are all 1 and σ_ℓ is a hyperparameter. In this definition, the importance of all features is equal and hyper-parameter σ_ℓ tunes the kernel's sensitivity to feature values. This definition is also advantageous in that σ_ℓ can automatically be tuned only using the training data [10]. We use this definition in our experiments.

3.1. Prediction of a single unseen datum

An advantage of the GP is that we do not need to perform numerical integration to calculate (1). Given the characteristics of Gaussian functions, y_* in (1) can be obtained analytically as follows where $\mathcal{N}$ denotes the *Gaussian (Normal) probability distribution*.

$$y_* \sim \mathcal{N}\left(\mathbf{k}_*^\top (K+\sigma_n^2 I)^{-1}\mathbf{y}, k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (K+\sigma_n^2 I)^{-1}\mathbf{k}_*\right) \quad (4)$$

In (4), $\mathbf{y} = (y_1, \ldots, y_M)$, $\mathbf{k}_* = (k(\mathbf{x}_*, \mathbf{x}_1), k(\mathbf{x}_*, \mathbf{x}_2), \ldots, k(\mathbf{x}_*, \mathbf{x}_M))^\top$, and K is an $M \times M$ matrix whose i, j element is defined as $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$.

In summary, given an unseen test sample $\mathbf{x}_*$, we can obtain its prediction using (4).

The GP is also advantageous in that hyperparameter optimization is computationally easy because of the use of the Gaussian function. To this point, we have the following hyperparameters: σ_f, σ_n, and $\mathbf{a}$. These hyperparameters can be tuned automatically so that the likelihood of $\mathcal{D}$ can be maximized.

3.2. Prediction of multiple unseen data

Section 3.1 discussed the prediction of a single unseen data $\mathbf{x}_*$. When n multiple unseen data, e.g., $\mathbf{x}_{*1}, \mathbf{x}_{*2}, \ldots, \mathbf{x}_{*n}$, the GP considers not only the closeness between each unseen data point and the training data but also the closeness between each unseen data point. In this case, the prediction can be written as follows.

$$\mathbf{y}_* \sim \mathcal{N}(\mu, \Sigma) \quad (5)$$

[3] $\top$ denotes the transpose of a vector or a matrix.

Here $\mu = (\mu_1, \ldots, \mu_n)^\top$ and Σ are the quality prediction and its covariance matrix, respectively. These play a key role and are used in the subsequent distribution process. They can be calculated analytically as follows.

$$\mu = K_*(K+\sigma_n^2 I)^{-1}\mathbf{y} \quad (6)$$

$$\Sigma = (K_{**}+\sigma_n^2 I) - K_*(K+\sigma_n^2 I)^{-1}K_*^\top \quad (7)$$

Here K_* is an $n \times M$ matrix whose i, j-th element is defined as $(K_*)_{i,j} = k(\mathbf{x}_{*i}, \mathbf{x}_j)$, and K_{**} is an $n \times n$ matrix whose i, j-th element is defined as $(K_{**})_{i,j} = k(\mathbf{x}_{*i}, \mathbf{x}_{*j})$.

In summary, given multiple unseen data points $\mathbf{x}_{*1}, \ldots, \mathbf{x}_{*n}$ as input, the GP outputs quality predictions in the form of a vector, $\mu = (\mu_1, \ldots, \mu_n)^\top$, and the (co-)variance matrix between the predicted values, Σ. Intuitively, the diagonal element of Σ, i.e., i, i-th element, encodes the risk or uncertainty of the prediction of the i-th unseen input. In addition, the nondiagonal element of Σ, i.e., the i, j-th element where $i \neq j$, encodes how uncertain the i-th prediction is when the j-th prediction is uncertain (and vice versa).

The theoretical background of the GP has been addressed in [10] . For implementation, we used the GPy toolkit [4], a GP library for the Python language.

4. Risk-aware Distribution of Translation Candidates

This section explains the key idea of the proposed approach: how the proposed system distributes translation candidates to readers considering the risk in translation quality. Assume that an SMT system outputs n-best translations for a source sentence. Here let $\mathbf{x}_{*1}, \mathbf{x}_{*2}, \ldots, \mathbf{x}_{*n}$ be the feature vectors constructed from the source sentence and the n-best translations. As explained in (3.2), given $\mathbf{x}_{*1}, \ldots, \mathbf{x}_{*n}$ as input, the GP outputs the predicted quality $\mu = (\mu_1, \ldots, \mu_n)^\top$ and the covariance matrix of the prediction Σ, which can be interpreted as the risk encoding how inaccurate the predicted quality might be.

Given μ and Σ, our goal is to calculate the *rate vector* $\lambda = (\lambda_1, \ldots, \lambda_n)^\top$ where each λ_i is the probability that i-th best translation is selected and distributed to a reader. In other words, λ_i determines what percentage of the entire readers should read the i-th best translation. This can be formally expressed as $\sum_{i=1}^n \lambda_i = 1$, and for each $i \in \{1, \ldots, n\}$, $\lambda_i \geq 0$.

The rate vector can be calculated by optimization using the following formula.

$$\text{maximize}_{\lambda_1,\ldots,\lambda_n} \quad \sum_{i=1}^n \lambda_i\mu_i - \frac{1}{2}\alpha\sum_{i=1}^n\sum_{j=1}^n \lambda_i\lambda_j(\Sigma)_{i,j} \quad (8)$$

$$\text{subject to} \quad \sum_{i=1}^n \lambda_i = 1 \quad (9)$$

$$\forall i \in \{1, \ldots, n\}, \lambda_i \geq 0 \quad (10)$$

[4] http://sheffieldml.github.io/GPy/

In (8), the objective function, i.e., the first term, attempts to maximize the predicted quality averaged over n candidates. In contrast, the second term penalizes the first term when the risk of the quality is large. Thus, (8) can be intuitively interpreted as maximizing the averaged predicted quality while penalizing the candidate whose risk is large. (8) has a hyperparameter, i.e., α, that tunes the strength of the risk penalization.

As explained, the constraints (9) and (10) guarantee that λ is always a probability vector whose elements can be interpreted as probability mass.

Notably, (8) includes the case wherein the predicted best translation is distributed to all readers. This case arises when we set α to 0. In this case, only the first term remains in (8). Because of the constraints (9) and (10), λ remains a probability vector in this case. Because of the first term that maximizes the quality, λ becomes a unit vector such that the i-th element with the highest μ_i value is set to 1 and all other elements are set to 0.

The solution of (8) can be obtained in practical time. Theoretically, (8) can be solved using linear-constrained convex optimization techniques, which obtain a global optimum. Moreover, through preliminary experiments, we found that we could find the solution in practical time. We were able to achieve good performance in average translation quality with small n, e.g., 5 and 3. In contrast, large n values degrade performance. This is presumably because n is the number of n-best translations output from SMT systems, and we rerank these outputs. Thus, n values that are too large increase the number of low-quality candidates and makes it difficult to determine good candidates.

After calculating λ, according to this vector, the proposed system distributes candidate translation to the readers.

5. Experiment Setup

We performed our experiments under two settings, i.e., a system selection setting and an n-best output setting. The n-best output setting is identical to what we have explained so far. Under the system selection setting, we use the n single-best outputs from n SMT systems as input rather than the n-best outputs of an SMT system. The system is required to distribute the n single best outputs to readers.

In both system selection and n-best output settings, we have simulated a case wherein translations are distributed to $1,000$ readers. In both settings, five-fold cross validation was performed. To extract features from the source text and translations, we used a standard QE system, *QuEST* [5].

For features, we used the basic 17 feature set defined in the literature [5]. Here, LM denotes a language model.

- Number of tokens in the source sentence
- Number of tokens in the target sentence
- Average source token length
- LM probability of source sentence
- LM probability of target sentence
- Number of occurrences of the target word within the target hypothesis
- Average number of translations per source word in the sentence
- Average number of translations per source word in the sentence weighted by the inverse frequency of each word in the source corpus
- Percentage of unigrams in quartile 1 of frequency, i.e., lower frequency words, in a corpus of the source language
- Percentage of unigrams in quartile 4 of frequency, i.e., higher frequency words, in a corpus of the source sentence
- Percentage of bigrams in quartile 1 of frequency of source words in a corpus of the source language
- Percentage of bigrams in quartile 4 of frequency of source words in a corpus of the source language
- Percentage of trigrams in quartile 1 of frequency of source words in a corpus of the source language
- Percentage of trigrams in quartile 4 of frequency of source words in a corpus of the source language
- Percentage of unigrams in the source sentence seen in a corpus
- Number of punctuation marks in the source sentence
- Number of punctuation marks in the target sentence

6. Quantitative Evaluation

6.1. Evaluation under system selection setting

For the system selection setting, we used the dataset from the system selection competition provided by WMT-13 quality estimation shared tasks [6]. This dataset uses an English-to-Spanish translation setting. Here we have five single best Spanish translations from five systems for an English source sentence. The proposed systems distribute these five Spanish translations to the readers.

In this dataset, through manual evaluation, it is known that the "online-B" system achieves the best translation quality. Thus, as a baseline, we considered a case wherein translation by the "online-B" system is given to all readers. In this dataset, 39.51% of translation by the "online-B" system were the actual best.

6.2. Compared methods

We also experimented with other methods for comparison. QE-max is a case where the best candidate with regard to QE score is given to all readers. Support vector regression-radial-basis function (SVR-RBF) is identical to QE-max, except that the quality prediction is calculated using SVR, a regression method based on a SVM, with an RBF kernel.

[5] http://staffwww.dcs.shef.ac.uk/people/L.Specia/projects/quest.html

[6] http://statmt.org/wmt13/quality-estimation-task.html

Table 5: Evaluation under System Selection Setting (values are sentence-level BLEU scores

Proposed	35.52
QE-max	35.43
SVR-RBF	34.98
Baseline	34.88

6.3. Hyperparameter tuning

Essentially we chose hyperparameters from nine points ranging in the log-space from 10^{-3} to 10^{3}: $10^{-12/4}, 10^{-9/4}, 10^{-6/4}, 10^{-3/4}, 10^{0}, 10^{3/4}, 10^{6/4}, 10^{9/4}$, and $10^{12/4}$.

For the GP, we used automatic tuning of hyperparameters with the training data [10], which is implemented in the GPy toolkit. Thus, the only hyperparameter that we tuned was α (Section 4), which tunes the strength of the risk penalization.

SVM-based regression with an RBF kernel has hyperparameters, i.e., C and γ. We chose C from these parameters. We fixed γ to 1 in this experiment.

6.4. Evaluation metric

Unlike previous studies, our objective is to improve the total quality of translations distributed to readers rather than improve the quality of the single best translation. Since previous studies did not focus on the number of readers, to the best of our knowledge, no previous evaluation metric specific to this situation has been proposed. This is problematic because previous evaluation metrics were not designed to take multiple translations as input although they are designed to handle multiple references.

For the evaluation, we simply interpreted the average of the quality scores passed to each reader as the metric for our evaluation. Even though no metric has been previously proposed for many readers, evaluation metrics for a single best translation have been studied extensively. We can evaluate the quality of the translation passed to one reader using an evaluation metric for a single best translation. By considering previously proposed metrics for a single best translation as metrics for a reader, it is natural to define the total quality of all readers as the quality score averaged over all readers. Moreover, these metrics for a single best translation have been tested extensively [3]. Therefore, we can leverage previous knowledge about these measures when analyzing our results.

For the actual evaluation metric for a reader, we have used sentence-level BLEU [3], because it is widely used for automatic evaluation when reference translations are available. For the implementation of sentence-BLEU, we used the "sentence-bleu" command bundled with the Moses toolkit.

6.5. Results

Table 5 shows our results. As can be seen, **Proposed** achieved the best results. We have also confirmed that **Proposed** significantly outperforms **Baseline**.

We also performed a Wilcoxon significance test for these results. As a result, **Proposed** was statistically significant against the **Baseline** ($p < 0.01$) and **QE-max** ($p < 0.01$).

Table 6: Evaluation under n-best Setting

Proposed	26.24
QE-max	26.06
Baseline	26.06

6.6. Evaluation under n-best setting

Here we evaluate the proposed approach in the n-best setting where n-best outputs from one SMT system are distributed to readers. For the SMT system, we used the English-to-Spanish translation setting so that we could use the same feature set as the system selection setting.

In this evaluation, we used the News Commentary corpus [7] so that the choice of corpus matches our task's target, i.e., web documents and news. The News Commentary corpus is a parallel corpus that comprises "news text and commentaries from the Project Syndicate." This corpus is provided as a part of the corpora for the series of WMT translation shared tasks.

We used the Moses toolkit trained with the News Commentary corpus as the SMT translator in our task. As usual for SMT evaluation, Minimum Error Rate Training (MERT) [14] was used to train the SMT translator. We used the same language pair, i.e., English-to-Spanish, for this evaluation, because a well-studied feature extractor for QE is provided for this language pair.

We set $n = 5$ in this experiment because, through a preliminary experiment, we found that it is quite rare for candidates ranked below fifth to be the actual best candidate. Indeed, in this experiment, only 34.23% of the first-ranked candidate was the actual best. The values for the second, third, fourth, and fifth ranked candidates were 21.31%, 17.05%, 13.92%, and 13.49%, respectively. The definitions of Baseline, QE-max, and Proposed are the same as those in Section 6.2.

Table 6 shows the results. Again, the proposed method clearly outperforms the other three methods. We also found statistical significance between **Baseline** and **Proposed** ($p < 0.01$).

7. Qualitative Evaluation by Examples

This section explains how the proposed method works successfully by demonstrating examples taken from **Proposed** in Section 6.6.

[7] http://www.statmt.org/wmt13/translation-task.html#download

Table 7: Two-top Example (the first two among the 5-best outputs are significantly better than latter cases)

Legend	Content	Actual BLEU	Predicted BLEU	Rate
Source text	Damascus, however , also brushed off this proposal .	-	-	-
1st best	Damasco , sin embargo , tambin desdeñó los esta propuesta .	23.46	27.74	0.45
2nd best	Damasco , sin embargo , tambin descartaron de esta propuesta .	23.46	27.74	0.55
3rd best	Damasco , sin embargo , tambin desdeñó los esa propuesta .	17.03	27.43	$< 10^{-6}$
4th best	Damasco , sin embargo , tambin desdeñó los de esta propuesta .	21.40	27.27	$< 10^{-6}$
5th best	Damasco , sin embargo , tambin los desdeñó los esta propuesta .	21.40	27.16	$< 10^{-6}$
Reference	entretanto , Damaskus critica tambin esta propuesta .	-	-	-

As mentioned previously, Table 7 shows the first example, which we call the "Two-top example."

By focusing on the first two elements in the **Actual BLEU** scores column, we can see that the actual BLEU scores of these elements are equal and are the highest among the five output translations. Since we cannot know the actual BLEU scores in advance, distributing only the "1st best" translation to all readers is risky because the "2nd best" might have a higher BLEU score. Thus, correctly recognizing these equal scores is crucial for handling this example.

The **Predicted BLEU** column shows the predicted BLEU scores obtained by GP-based quality estimation, i.e., the elements of the vector μ (Section 3). Comparing the predicted and actual BLEU scores, we find that the predicted values are not particularly accurate. The actual BLEU scores for all five examples are < 24; however, all of the predicted scores are > 27. The reason for this is presumably because the reference translation in this example is structurally different from the source text and its translation candidates, i.e., "however" in the source sentence is placed in the middle of the sentence as an adverb, and in the reference translation, the conjunction "entretanto" (meanwhile) is used instead and is placed at the beginning of the sentence. This result clearly demonstrates the difficulty of accurately estimating an exact value for the BLEU scores. Although actual BLEU scores depend on the reference translations, in QE, we must estimate the scores without reference translations.

Although the **Predicted BLEU** scores in Table 7 are not accurate as a regression problem, these scores successfully capture the overall characteristics in the order of the candidates with regard to their quality in this example. The first two are significantly better than the rest. Thus, we can see that the **Predicted BLEU** scores can be leveraged if we use the scores intelligently.

In the fifth column, the **Rate** vector, which we define in Section 4, successfully captures the basic characteristics of the five candidates because of the use of the (co-)variance matrix. The first two candidates consume nearly all of the weights that are to be sum up to 1.0. The rates for the latter three candidates are $< 10^{-6}$, which indicates that these candidates are almost ignored and are essentially never distributed to readers. This reflects the fact that the two top candidates are by far better than the latter candidates. We also find that the probability allocated to the first two candidates is close to 0.5. This implies that our risk-aware distribution system successfully recognizes that the first two candidates are scored equally, and this decision is reflected in the rate vector.

In summary, these experimental results show that our distribution system correctly recognizes that the first two candidates are significantly better than the latter cadidates and that they are scored equally. Thus, our system distributes the first two translations considering the case in which the second best translation would actually be better than the first. In this example, since the actual BLEU scores of the first two candidates are equal, the quality is not improved compared to the case wherein the "1st best" is distributed to all readers. However, if the actual BLEU score of "1st best" was even slightly less than that of "2nd best," our approach would have successfully outperformed the baseline.

8. Discussion

The optimization problem used to determine the rate of distribution introduced in Section 4 is a type of *multi-objective optimization*. In multi-objective optimization, there are multiple objective functions to optimize, and the goal is to optimize the functions simultaneously. In our application, we simultaneously maximize the predicted quality of the translations distributed to readers while minimizing risks. This use of multi-objective optimization is based on modern-portfolio theory, where the goal is to maximize financial profit rather than translation quality [15]. However, our task is more than a simple application of modern-portfolio theory in that we cannot directly measure the objective function and its variances, whereas these are assumed to be directly observable in modern portfolio theory. This unavailability of direct measurement of the objective function and its variances is the reason why we predict it from the training data using GP-based QE (Section 3).

Unlike our task, previous use of multi-objective optimization in machine translation studies appears limited to simultaneously optimizing multiple evaluation metrics. A previous study [16] used multi-objective optimization to optimize multiple automatic evaluation metrics simultaneously,

i.e., BLEU and RIBES [17]. Another study used multi-objective optimization to optimize document-level evaluation metrics and sentence-level evaluation metrics [18]. In computational linguistics, other than machine translation tasks, multi-objective optimization was recently used in joint disambiguation of nouns and named entities [19].

9. Conclusion

In this paper, we have proposed an approach for distributing translation candidates to readers for translated documents with many anonymous readers, such as web documents and news articles. Our key idea is to use all translation candidates rather than the top candidate in consideration of the risk that the top candidate actually has lower quality than other candidates. Our experimental results show that the proposed approach consistently outperforms the baseline approach wherein the top candidate is distributed to all readers.

In future, we would like to test the proposed approach with other language pairs.

10. References

[1] P. Koehn, *Statistical machine translation*. Cambridge University Press, 2009.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proc. of ACL interactive poster and demonstration sessions*, 2007, pp. 177–180.

[3] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. of ACL*, Philadelphia, Pennsylvania, USA, July 2002, pp. 311–318.

[4] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing, "Confidence estimation for machine translation," in *Proc. of COLING*, Geneva, Switzerland, Aug 23–Aug 27 2004, pp. 315–321.

[5] L. Specia, N. Cancedda, M. Dymetman, M. Turchi, and N. Cristianini, "Estimating the sentence-level quality of machine translation systems," in *Proc. of EAMT*, 2009, pp. 28–37.

[6] T. Cohn and L. Specia, "Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation," in *Proc. of ACL*, Sofia, Bulgaria, August 2013, pp. 32–42.

[7] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, M. Huck, C. Hokamp, P. Koehn, V. Logacheva, C. Monz, M. Negri, M. Post, C. Scarton, L. Specia, and M. Turchi, "Findings of the 2015 workshop on statistical machine translation," in *Proc. of WMT*, Lisbon, Portugal, September 2015, pp. 1–46.

[8] D. Beck, K. Shah, T. Cohn, and L. Specia, "SHEF-Lite: When less is more for translation quality estimation," in *Proc. of WMT*, Sofia, Bulgaria, August 2013, pp. 337–342.

[9] D. Beck, K. Shah, and L. Specia, "Shef-lite 2.0: Sparse multi-task gaussian processes for translation quality estimation," in *Proc. of WMT*, Baltimore, Maryland, USA, June 2014, pp. 307–312.

[10] C. Williams and C. Rasmussen, *Gaussian processes for machine learning*. MIT Press, 2006.

[11] O. Bojar, M. Ercegovčević, M. Popel, and O. Zaidan, "A grain of salt for the wmt manual evaluation," in *Proc. of WMT*, Edinburgh, Scotland, July 2011, pp. 1–11.

[12] K. Heafield and A. Lavie, "Cmu system combination in wmt 2011," in *Proc. of WMT*, Edinburgh, Scotland, July 2011, pp. 145–151.

[13] S. Kumar and W. Byrne, "Minimum bayes-risk decoding for statistical machine translation," in *Proc. of HLT-NAACL*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 169–176.

[14] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, 2003, pp. 160–167.

[15] H. Markowitz, "Portfolio selection*," *The journal of finance*, vol. 7, no. 1, pp. 77–91, 1952.

[16] K. Duh, K. Sudoh, X. Wu, H. Tsukada, and M. Nagata, "Learning to translate with multiple objectives," in *Proc. of ACL*, Jeju Island, Korea, July 2012, pp. 1–10.

[17] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada, "Automatic evaluation of translation quality for distant language pairs," in *Proc. of EMNLP*, Cambridge, MA, October 2010, pp. 944–952.

[18] C. Ding, M. Utiyama, and E. Sumita, "Document-level re-ranking with soft lexical and semantic features for statistical machine translation," in *Proc. of AMTA*, 2014.

[19] D. Weissenborn, L. Hennig, F. Xu, and H. Uszkoreit, "Multi-objective optimization for the joint disambiguation of nouns and named entities," in *Proc. of ACL-IJCNLP*, Beijing, China, July 2015, pp. 596–605.

Inducing Bilingual Lexicons from Small Quantities of Sentence-Aligned Phonemic Transcriptions

Oliver Adams[1], Graham Neubig[2], Trevor Cohn[1], Steven Bird[1]

[1]Department of Computing and Information Systems, The University of Melbourne, Australia
[2]Graduate School of Information Science, Nara Institute of Science and Technology, Japan
oadams@student.unimelb.edu.au, neubig@is.naist.jp, {t.cohn,sbird}@unimelb.edu.au

Abstract

We investigate induction of a bilingual lexicon from a corpus of phonemic transcriptions that have been sentence-aligned with English translations. We evaluate existing models that have been used for this purpose and report on two additional models, which demonstrate performance improvements. The first performs monolingual segmentation followed by alignment, while the second performs both tasks jointly. We show that monolingual and bilingual lexical entries can be learnt with high precision from corpora having just 1k–10k sentences. We explain how our results support the application of alignment algorithms to the task of documenting endangered languages.

1. Introduction

Language documentation involves the construction of text collections, lexicons and grammars in the interest of creating a record of a language for future linguistic, cultural and anthropological analysis. Traditional approaches to language documentation are labour-intensive, requiring much one-on-one time between a field linguist and the mother tongue speakers. Unfortunately, there aren't enough linguists to document the world's languages using these approaches before many of the approximately 7,000 languages die out.

There is a movement to increase the rate of data collection of endangered languages using cheap and widespread electronics to record speech in a more ad hoc manner [1, 2, 3, 4, 5], in an attempt to provide the field linguist with leverage to acquire data faster. This data is primarily audio, since most languages have no established written form and capturing audio is comparatively fast. Additionally, much of the data is bilingual, as an important aspect of the language documentation process is the construction of bilingual corpora and lexicons.

In this paper we consider the task of automatically learning monolingual and bilingual lexical items from unsegmented phonemic transcriptions of interleaved audio (segments of speech in one language along with spoken translations in another). Such transcriptions could arise from two scenarios. The first is when future philologists phonetically transcribe speech of a language post-mortem, without native speakers to assist in word segmentation. In such instances lexicon induction would aid in linguistic analysis of the language. The second is by instead employing automatic speech recognition technologies for the same task. In both cases lexicon induction could aid in bootstrapping automatic speech recognition (ASR) systems targeting the language's untranscribed audio. Note that we assume a transcription of the English translation, since English speech can be reliably and cheaply transcribed.

Previous work on bilingual lexicon induction using sentence-aligned corpora has focused primarily on large corpora of written text [6, 7, 8, 9]. However, bilingual lexicon induction applied to phonemically transcribed audio introduces problems, including the lack of word segmentation and the small quantities of data. There has been limited work on learning lexicons from phonemic transcriptions. [10, 11] take a first look at phoneme–word translation modeling, using traditional IBM Models [12] in order to determine alignments and applying heuristics to extract dictionaries. [13] propose Model 3P, which builds upon the generative story of IBM Model 3 by adding additional word length parameters and allowing it to significantly outperform the IBM models [14, 15, 16].

Building on this work, we investigate two models that haven't been considered in this context and demonstrate that they can outperform the models that have been considered. The first performs unsupervised word segmentation followed by word alignment. The second jointly performs word segmentation and alignment. Importantly, we evaluate the models on a data set that is significantly smaller than has been evaluated on previously, containing between just 1k and 10k sentences, corresponding to 13k and 132k words. This likely corresponds to something in the order of 1 to 10 hours of speech [17, 18, 5]. These quantities of data are realistic in the context of documentation of endangered languages, though the applicability of these techniques also applies more generally to low-resource languages that have no body of written resources.

We run experiments to assess the induced lexicons' precisions at k entries. We do this by applying the alignment models to a German–English corpus, using heuristics to ex-

tract lexical entries before having them manually annotated[1].

German was used since it permitted easier manual annotation of lexical entries than an endangered language. Although German and English are more closely related languages than language pairs encountered in linguistic fieldwork, modeling of the language pair is still complex due to varying word order between the languages and the morphological richness of German relative to English.

Results demonstrate that hundreds of bilingual lexical entries can be learnt with good precision, with the additional proposed methods outperforming Model 3P on a data set of 10k sentences. This offers promise of the technique's applicability in a language documentation context. Moreover, the majority of incorrect entries correspond to well-segmented, but misaligned, source words.

2. Translation Models

Our lexicon induction approach uses various phrase alignment techniques to segment sequences of phonemes into words and learn phrase tables. There are several methods for word segmentation in machine translation [19, 20, 21, 22, 16], but there has been limited application in a low resource context. In this paper we examine four representative methods to apply to parallel sentences comprised of source phoneme tokens and target words.

The first two, GIZA++ and Model 3P, have been investigated previously for the task of phoneme–word alignment [10, 14]. They are evaluated as a point of comparison for the latter two methods we demonstrate are effective for this task, which use unsupervised word segmentation (UWS) with GIZA++ and a Bayesian inversion transduction grammar (ITG) framework.

2.1. GIZA++

GIZA++ is the baseline that follows the standard statistical machine translation (SMT) pipeline of performing alignment with the IBM Models [12], as implemented in GIZA++ [23]. This approach to alignment was used in seminal work on phoneme–word alignment [10, 11]. The problem with this approach is that it attempts to capture relationships between individual foreign phonemes and English words, which is extremely difficult.

2.2. Model 3P

PISA[2] is an implementation of the Model 3P model of [13]. It builds upon the generative model of IBM Model 3 [12] by adding additional word length parameters (see Figure 1), allowing it to outperform traditional IBM models on phoneme–word alignment tasks. After initializing model parameters with learnt GIZA++ parameters, the PISA implementation

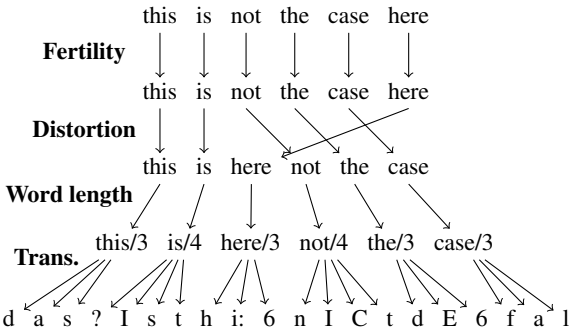


Figure 1: The generative model of Model 3P.

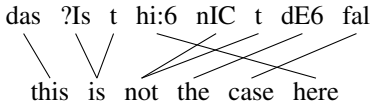


Figure 2: Monolingual segmentation of phonemes followed by alignment, as done in the UWS GIZA++ approach.

of Model 3P uses a genetic algorithm to learn the parameters of the model.

The additional word length parameters, distinct from the fertility parameters, allow Model 3P to learn latent word representations that would not be able to be captured in a direct phoneme–word mapping. This allows for better segmentation performance.

2.3. UWS GIZA++

UWS GIZA++ first performs unsupervised word segmentation using the Bayesian Pitman-Yor language model [24], as implemented in the tool pgibbs[3] [25]. Alignment is then performed between these phoneme sequences and the English words using GIZA++ (see Figure 2). This was hypothesized to be more appropriate than GIZA++ alone since it would result in breaking the foreign phoneme sequences into coarser tokens that translate better to English. Note that there is not an expectation that the word segmentation perform well with respect to what is considered a "word" in the given language. Instead, the key idea is that the segmenter breaks phonemes into frequently repeating units that capture more meaning than just using individual phonemes. Consider Figure 2: the erroneous segmentation nevertheless allows for accurate alignment after monolingual segmentation.

2.4. Bayes ITG

Bayes ITG performs joint word segmentation and alignment using the substring alignment model of [26], as implemented in pialign[4] [27]. Alignments are obtained through Bayesian learning of inversion transduction grammar trees [28], which

[1]These annotations will be released along with code for the lexicon induction.

[2]https://code.google.com/p/pisa

[3]http://github.com/neubig/pgibbs

[4]http://github.com/neubig/pialign

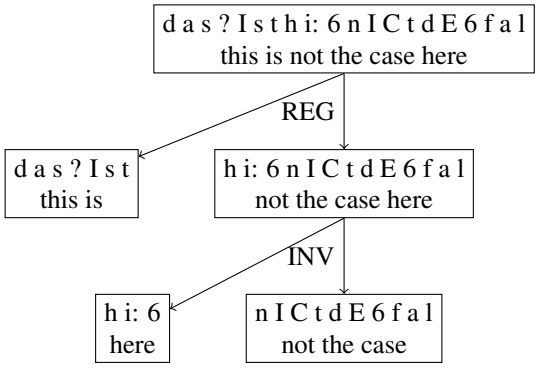


Figure 3: An ITG tree structure learnt by pialign. Note that pialign forces alignments down to individual tokens, but the leaf nodes presented here represent alignments that were generated as single phrase by the model.

completely describe the sentence and its translation as a tree of aligned phrases and binary reordering operations. In Figure 3 the sentence is decomposed, with phrases of different granularities being captured. The *REG* and *INV* tags illustrate the reordering capacities of the ITG trees, with *REG* being a monotone alignment ordering and *INV* flipping the English side with respect to the foreign phonemes. The advantage of this joint learning approach over GIZA++, Model 3P and UWS GIZA++ is that the segmentation on the phoneme side can be informed by the English, which has been shown to be valuable [20, 21, 22]. Furthermore, the base distribution Bayes ITG draws from uses cooccurrence probabilities of phrases. This contrasts with Model 3P's initialization, which uses only the limited phoneme–word alignments of GIZA++.

3. Experimental Setup

3.1. Data

To train the translation models we used the German–English parallel corpus from Europarl v7 [29]. In order to imitate a phoneme transcription, we converted the German side to a sequence of phonemes (represented with the SAMPA[5] phoneme alphabet) using the MARY text-to-speech system [30]. For example, 'dieser' is represented as a sequence of space-separated phonemes, 'd i: z 6'.

The phonemic output of MARY includes some information that cannot reasonably be detected by an ASR system. In particular, stress markers and syllable boundaries are features output by the system ('ˈ' and '-' respectively), so we filtered them out. The granularity of tokens on the source side was thus at the phoneme level while English words were used on the target side.

Small quantities of data were used in order to mimic the realities of data collection for endangered languages. We experimented with varying data sizes to evaluate how the best method's performance scales. We used data sets of 1k, 2k, 5k, and 10k parallel sentences (corresponding to between ~13k and ~132k words), a quantity that is vastly smaller than what is typically used in statistical machine translation experiments but which approaches reasonable size for reliable manual transcription. We limited training sentences to those fewer than 100 phonemes in length.

[5] http://www.phon.ucl.ac.uk/home/sampa/german.htm

3.2. Translation Model Training Parameters

GIZA++ was trained using the train-model.perl script included in Moses with default settings, using the grow-diag-final-and heuristic for symmetrization/phrase extraction and the msd-bidirectional-fe reordering model.

PISA was trained with default settings.

UWS GIZA++ was trained by running pgibbs first, and then running GIZA++ over the segmented phoneme sequences with default settings. The pgibbs settings were default, with the following exceptions: block sampling was used with a block size of 50, a Pitman-Yor distribution was used, and 1000 iterations were run. The final sample output by pgibbs was used as input to GIZA++. GIZA++ was run in the same way as above, using train-model.perl with heuristics for phrase extraction. It's worth noting that the hyperparameters supplied to pgibbs dictate segmentation granularity. Were they to change, we would expect the average length of the word units learnt to be different.

We ran pialign for 10 iterations with the base distribution being a log-linear interpolation of phrase cooccurrence probabilities in both directions (with a discount of 5), a beam width of 10^{-6} and a batch length of 40. The final sample was used for the purposes of phrase table extraction.

3.3. Bilingual Lexicon Extraction

To create bilingual lexicons using the above approaches, entries in the phrase tables were first sorted according to their joint probabilities. We only included entries where the length of the phonemic side was 2 or greater. This heuristic was used since it removed many spurious entries where one foreign phoneme was aligned to an entire word. Additionally, for a given English entry no more than the top 5 translations were included. A similar filter was applied to prevent more than 5 English translations of a given phoneme sequence. The top 500 entries of each lexicon were then manually annotated.

3.4. Annotation

Entries in the lexicon were evaluated by a native German speaker.[6] They were determined to be correct, incorrect or ambiguous. Correct entries are those that can readily be found in existing German–English dictionaries. For example, the entry *vIs@n⇔know* ('wissen'). Incorrect entries are those whose translations are deemed to be clearly incorrect

[6] We measured inter-annotator agreement by doubly annotating a sample of 1k entries, using a non-native German speaker, resulting in $\kappa = 0.69$.

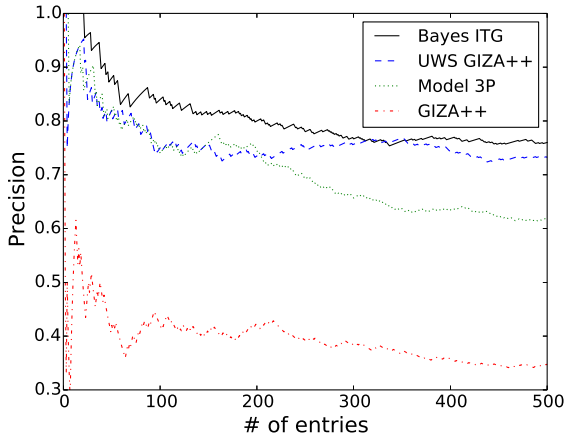


Figure 4: Comparison of the methods' precisions over the 10k dataset. Note that these are the results of the strict evaluation.

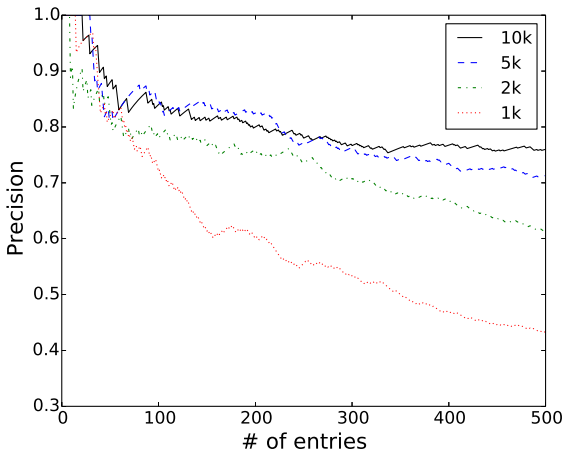


Figure 5: Comparison of Bayes ITG precisions for different sized data.

by the annotator. These include entries such as *tsu:?aIn⇔the* and *b@dINUN⇔be*. In the latter case, note that although the word alignment is incorrect, the phonemes represent a correctly segmented German word, 'Bedingung'.

Ambiguous entries are those that are neither strictly correct nor incorrect. These include entries that have boundary errors. For example, *nvi:6⇔we* ('wir') includes an extra 'n' in an otherwise correct entry. Other ambiguous entries are those that, while not found in lexicons, are nonetheless meaningful. These usually highlight interesting linguistic phenomena. For example, *nICt⇔does not* ('nicht') couldn't be found in Leo,[7] however it captures a meaningful grammatical relationship between the languages. Consider the phrase 'er rennt nicht' and one English translation 'he does not run', where this entry makes sense.

4. Quantitative Evaluation

4.1. Precision at k over bilingual entries

We compare the four models described in Section 2, each of which takes as input sentences of unsegmented phonemes and English translations. Figure 4 shows the precisions of the bilingual lexicons as the number of entries increases from 1 to 500 (sorted by the joint probability given by the model), using the methods trained on 10k sentences.[8] The 'traditional' approach with GIZA++ is the worst performer across the board. This is to be expected as it uses lexical translation probabilities between poorly translated German phonemes and English words as the basis for the extracted phrases. As a point of comparison to these models, we trained an 'oracle' model on correctly segmented phonemes using GIZA++, which removes the effect of segmentation errors but still includes the effect of alignment errors. This oracle model yielded a lexicon with a precision of 0.932 over the top 500 entries.

The other methods are more similar in performance, with the best performing approach being Bayes ITG. Though the results are close, the better performance of Bayes ITG as compared to the unsupervised word segmentation approach can possibly be attributed to the added information the English side provides in determining useful German phrases. This contrasts to the unsupervised word segmentation approach which segments using only monolingual German phonemic data. Performance gains over PISA's Model 3P can perhaps be attributed to limitations in Model 3P's generative model. Rather than learning explicit phrasal relationships between phoneme groups and words, Model 3P conditions the generation of phonemes from latent words and the location within that word.

Similar trends in the scores were demonstrated when evaluating precisions that accepted ambiguous entries as also correct.

Given that Bayes ITG was the best-performing approach on 10k sentences, we additionally evaluated it on smaller data sizes (see Figure 5). The fewer sentences of phonemes that are supplied the more reasonable it is to assume that they can be acquired through reliable manual transcription in a real language preservation scenario. Precision appears to be a logarithmic function of the size of the training data. These results suggest that the first few hundred entries in a lexicon can be acquired with good precision even with very limited data.

4.2. Word segmentation performance

In addition to evaluating the quality of the bilingual entries, we evaluated the quality of monolingual lexical entries on

[7] http://www.leo.org

[8] Note that we do not investigate recall as it is both difficult to establish and less relevant in the early stages of language documentation as only a small fraction of words will be captured in any case.

Method	Sents	Incorrect %	Correct seg. %
Bayes ITG	1k	26.2	52.7
Bayes ITG	2k	16.6	60.2
Bayes ITG	5k	13.4	62.7
Bayes ITG	10k	9.6	62.5
UWS GIZA++	10k	7.2	38.9
GIZA++	10k	19.4	15.5
Model 3P	10k	14.6	46.6

Table 1: The accuracy of the segmentation of phonemic lexical entries judged incorrect. The *Incorrect %* columns indicate the percentage of the 500 annotated entries that were labeled completely *incorrect* as bilingual entries . The *Correct seg. %* column indicates the percentage of those *incorrect* entries that were correctly segmented monolingual entries.

the phoneme side. This is motivated by the observation that often correct phonemic word units were extracted but mistranslated. Since monolingual entries are useful in their own right for language documentation purposes (for instance, as a useful starting point for manual correction) and language modeling, we assessed entries that were *incorrect* to determine whether the phonemic component was segmented correctly at the word boundaries.

Table 1 shows the proportion of the total entries that were annotated as *incorrect* and the proportion of those entries that were correct monolingual lexical entries on the phoneme side. Bayes ITG demonstrates effective inference of lexical items with few boundary errors, outperforming the other methods regardless of the amount of training data used. This corroborates past research that indicates that word segmentation can be better informed with bilingual data [20, 21, 22].

Also noteworthy is the outperformance of Model 3P relative to UWS GIZA++ when entries are *correct* (though having fewer strictly *incorrect* entries overall). In the approach of UWS GIZA++ it is impossible to break apart phoneme groups that have been chunked across word boundaries by the monolingual segmentation phase. However, the other methods aren't constrained by early, poorly informed chunking. This allows Model 3P relatively better word segmentation despite lower precision of bilingual lexical entries.

Note that although we are evaluating monolingual entries, the entries of UWS GIZA++ are still informed by the alignments with English, as the entries evaluated are the highest probability bilingual lexical entries found. This mitigates the problem of the effort required to tweak the hyperparameters of the word segmenter to find the right granularity of phoneme clusters. The granularity is instead informed by the English. To appreciate this, consider the most occurring lexical entries of the monolingual supervision *without* being informed by the alignments, as shown in Table 2. Of these, the only one that is an actual word is *di:* ('die'). The rest are common sub-word units. Note though that *@n* ('-en') is a common suffix for infinitive verbs—a particularly useful morpheme.

Token	Occurrences
?	13,096
@	8,587
n	8,138
t	6,422
@n	6,300
d	5,929
s	3,226
6	3,136
f	3,099
di:	2,913

Table 2: The most common lexical entries found by the unsupervised word segmentation, without harnessing bilingual information.

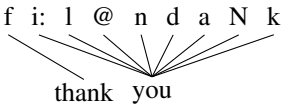


Figure 6: The phonemes of *vielen dank* as aligned to *thank you* by PISA's Model 3P.

5. Qualitative Evaluation

To appreciate the peculiarities and differences of these approaches, we will now consider some general observations made by examining the lexicons of the various approaches, discussing some representative lexical entries and word alignments.

Model 3P seemed generally more susceptible to off-by-one errors at the boundaries of entries. A high confidence, but incorrect, entry that occurred in the lexicon based on Model 3P alignments was *i:l@ndaNk⇔you* ('vielen dank'). The English makes some sense, as *vielen dank* can be translated as 'thank you' or 'thank you very much', although the 'thank' component on the English side is missing. Notably, the German side is segmented incorrectly at the phrase boundary, missing the initial phoneme 'f' (it should be 'fi:l@ndaNk'). It turns out that in sentences containing this German phoneme sequence, the 'f' is often aligned to English 'thank' (see Figure 6). In the lexicons created by both Bayes ITG and UWS GIZA++ this entry was correctly phrase-segmented as 'fi:l@ndaNk'.

A similar such entry in the Model 3P lexicon was *daspa6la:mEn⇔parliament*, where the source side is missing the final 't'. In the lexicon constructed using Bayes ITG, such boundary mistakes were scarce. The equivalent entry was *daspa6la:mEnt⇔parliament* ('das Parlament'). Note that this entry was not considered strictly correct nor correctly segmented, as it is comprised of two words, with the German article being included. However in this case, as in almost all others, Bayes ITG still segments correctly at the boundaries of multiword units (as distinct from correctly seg-

mented individual words). One of the instances of an entry annotated as *incorrect* in the top 500 entries of the Bayes ITG lexicon where the phoneme side was also incorrectly segmented was *tvo6d@n⇔been*, where there is a spurious '*t*' prefixing the phonemic representation of 'worden'. Investigating the alignments highlights the cause of this entry. Phoneme sequences such as *Unt6StYtstvO6d@n* ('unterstützt worden') and ?E6RaICtvO6d@n ('erreicht worden') include verbs that often appear inflected with different suffixes elsewhere, but end in 't' when occurring before *vO6d@n* ('unterstützen' and 'erreichen' respectively, with the suffix '-en'). High correlation of *vO6d@n* ('worden') and the suffix '*t*' likely caused this entry.

The lexicon constructed using Model 3P demonstrated an apparent bias to shorter units. In that lexicon, the above entry was segmented correctly as *vO6d@n*. On the other hand, Bayes ITG tended to learn towards longer multiword units, as a result of the model's capacity to capture phrases at coarser granularities. *po:Ete:6RIN⇔Mr Poettering* was present in the Bayes ITG lexicon, but not in the others. The title is missing on the source side. This can be attributed to varying morphology of the title, which takes the form of both 'Herr' and 'Herrn' depending on context. However, since the English side consistently takes the form of 'Mr Poettering', evidence is built up primarily to relate both the title and name on the English side to only the name on the phoneme side.

For all the alignment approaches, there were many entries that are justified given only the information present in the corpus. The above example, *daspa6la:mEn⇔parliament*, is one such example and is arguably correct in some contexts (consider the phrase *das Parlament lehnte den Antrag ab⇔Parliament rejected the request*). This entry can be attributed to linguistic differences that possibly no alignment algorithm can overcome, with the article often being optional in English translations. In general, the entries Bayes ITG presented us with tend to be interpretable with respect to how phoneme sequences occur in the corpus.

UWS GIZA++ yielded the high confidence, yet erroneous, entries *t?⇔is*, *n?⇔to*, *n?⇔of*, which didn't occur in the other lexicons. This is likely a result of the pipelined nature of the approach, where monolingual segmentation is first performed before alignment. The German components to these entries represent frequently occurring phonemic sequences (many words end with '*t*' or '*n*' and many start with a glottal stop, '*?*', before some vowel). The English sides represent function words that are so commonly occurring that the coincidental cooccurrence of these phonemes and English words allowed them to become extracted lexical entries, which were not obtained using Bayes ITG or Model 3P. Entries such as this partly explain why UWS GIZA++ failed to perform as well as Model 3P in segmenting lexical entries despite outperforming it in bilingual precision. The other likely reason is that chunks that cross word boundaries learnt during monolingual segmentation cannot be undone.

6. Conclusion

We compared four representative approaches, evaluating the quality of monolingual and bilingual lexical entries. While two of the techniques had been previously established for the task of phoneme–word alignment, we achieved performance improvements by applying models that had not previously been considered for this task, demonstrating that hundreds of bilingual lexicon entries can be learnt with as few as 1k sentences of bilingual data. This can be done despite using an unsegmented phonemic representation of the source side.

Such approaches may be used to indicate what can be inferred from corpora of interleaved audio in the absence of reliable segmentation, aid in post-mortem linguistic analysis of a language, and to bootstrap ASR systems in order to help improve their phoneme recognition.

7. References

[1] T. Hughes, K. Nakajima, L. Ha, A. Vasu, P. J. Moreno, and M. LeBeau, "Building transcribed speech corpora quickly and cheaply for many languages." in *INTERSPEECH*, 2010, pp. 1914–1917.

[2] N. J. De Vries, J. Badenhorst, M. H. Davel, E. Barnard, and A. De Waal, "Woefzela-an open-source platform for ASR data collection in the developing world," in *INTERSPEECH*, 2011.

[3] N. J. De Vries, M. H. Davel, J. Badenhorst, W. D. Basson, F. De Wet, E. Barnard, and A. De Waal, "A smartphone-based ASR data collection tool for under-resourced languages," *Speech communication*, vol. 56, pp. 119–131, 2014.

[4] D. W. Reiman, "Basic oral language documentation," in *Language Documentation & Conservation*. University of Hawai'i Press, December 2010, pp. 254–268.

[5] S. Bird, F. R. Hanke, O. Adams, and H. Lee, "Aikuma: A mobile app for collaborative language documentation," in *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*. Baltimore, Maryland, USA: ACL, June 2014, pp. 1–5.

[6] D. Wu and X. Xia, "Learning an English-Chinese lexicon from a parallel corpus," in *Proceedings of the First Conference of the Association for Machine Translation in the Americas*, 1994, pp. 206–213.

[7] I. D. Melamed, "Automatic construction of clean broad-coverage translation lexicons," *CoRR*, 1996.

[8] H. M. Caseli, V. N. Maria das Graças, and M. L. Forcada, "Automatic induction of bilingual resources from aligned parallel corpora: application to shallow-transfer machine translation," *Machine Translation*, vol. 20, no. 4, pp. 227–245, 2006.

[9] A. Lardilleux, J. Gosme, and Y. Lepage, "Bilingual lexicon induction: Effortless evaluation of word alignment tools and production of resources for improbable language pairs," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC-2010)*. Valletta, Malta: European Languages Resources Association (ELRA), May 2010, pp. 252–256.

[10] S. Stüker and A. Waibel, "Towards human translations guided language discovery for ASR systems." in *SLTU*, 2008, pp. 76–79.

[11] S. Stüker, L. Besacier, and A. Waibel, "Human translations guided language discovery for ASR systems." in *INTERSPEECH*, 2009, pp. 3023–3026.

[12] P. E. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–312, 1993.

[13] F. Stahlberg, T. Schlippe, S. Vogel, and T. Schultz, "Word segmentation through cross-lingual word-to-phoneme alignment," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 85–90.

[14] ——, "Pronunciation extraction from phoneme sequences through cross-lingual word-to-phoneme alignment," in *Statistical Language and Speech Processing*. Springer, 2013, pp. 260–272.

[15] ——, "Towards automatic speech recognition without pronunciation dictionary, transcribed speech and text resources in the target language using cross-lingual word-to-phoneme alignment," in *Spoken Language Technologies for Under-Resourced Languages*, 2014.

[16] ——, "Word segmentation and pronunciation extraction from phoneme sequences through cross-lingual word-to-phoneme alignment," *Computer Speech & Language*, pp. –, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230814000977

[17] C. Cieri and M. Liberman, "More data and tools for more languages and research areas: a progress report on ldc activities," in *5th International Conference on Language Resources and Evaluation (LREC2006), Genoa, Italy*, 2006.

[18] S. Bird and D. Chiang, "Machine translation for language preservation." in *24th International Conference on Computational Linguistics*, 2012, p. 125.

[19] Y. Deng and W. Byrne, "HMM word and phrase alignment for statistical machine translation," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: ACL, October 2005, pp. 169–176.

[20] J. Xu, J. Gao, K. Toutanova, and H. Ney, "Bayesian semi-supervised Chinese word segmentation for statistical machine translation," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. ACL, 2008, pp. 1017–1024.

[21] 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*. Columbus, Ohio: ACL, June 2008, pp. 224–232.

[22] T. Nguyen, S. Vogel, and N. A. Smith, "Nonparametric word segmentation for machine translation," in *Proceedings of the 23rd International Conference on Computational Linguistics*. ACL, 2010, pp. 815–823.

[23] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[24] D. Mochihashi, T. Yamada, and N. Ueda, "Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: ACL, August 2009, pp. 100–108.

[25] G. Neubig, "Simple, correct parallelization for blocked Gibbs sampling," Nara Institute of Science and Technology, Tech. Rep., 2014. [Online]. Available: http://www.phontron.com/paper/neubig14pgibbs.pdf

[26] G. Neubig, T. Watanabe, S. Mori, and T. Kawahara, "Machine translation without words through substring alignment," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jeju Island, Korea: ACL, July 2012, pp. 165–174.

[27] G. Neubig, T. Watanabe, E. Sumita, S. Mori, and T. Kawahara, "An unsupervised model for joint phrase alignment and extraction," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 632–641.

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

[29] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT Summit*, vol. 5, 2005, pp. 79–86.

[30] 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.

Index

iwslt.org