WMT 2018

# Third Conference on Machine Translation

## Proceedings of the Conference

October 31 - November 1, 2018
Brussels, Belgium

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-948087-81-0

# Introduction

The Third Conference on Machine Translation (WMT 2018) took place on Wednesday, October 31 and Thursday, November 1, 2018 in Brussels, Belgium, immediately preceding the Conference on Empirical Methods in Natural Language Processing (EMNLP 2018).

This is the third time WMT has been held as a conference. The first time WMT was held as a conference was at ACL 2016 in Berlin, Germany, and the second time was at EMNLP 2017 in Copenhagen, Denmark. Prior to being a conference, WMT was held 10 times as a workshop. WMT was held for the first time at HLT-NAACL 2006 in New York City, USA. In the following years the Workshop on Statistical Machine Translation was held at ACL 2007 in Prague, Czech Republic, ACL 2008, Columbus, Ohio, USA, EACL 2009 in Athens, Greece, ACL 2010 in Uppsala, Sweden, EMNLP 2011 in Edinburgh, Scotland, NAACL 2012 in Montreal, Canada, ACL 2013 in Sofia, Bulgaria, ACL 2014 in Baltimore, USA, and EMNLP 2015 in Lisbon, Portugal.

The focus of our conference is to bring together researchers from the area of machine translation and invite selected research papers to be presented at the conference.

Prior to the conference, in addition to soliciting relevant papers for review and possible presentation, we conducted 8 shared tasks. This consisted of three translation tasks: Machine Translation of News, Biomedical Translation, and Multimodal Machine Translation, two evaluation tasks: Metrics and Quality Estimation, as well as the Automatic Post-Editing and Parallel Corpus Filtering tasks. The Parallel Corpus Filtering tasks was run at this year's edition of WMT for the first time. As almost all machine translation system require parallel corpora to train their models, the size and quality of available parallel corpora has a substantial impact on machine translation quality. At the same, sizable, high-quality parallel corpora are not available for many languages. This task addresses the important issue of how to exploit noisy parallel corpora, which are available in much larger quantities and for a larger number of languages.

The results of all shared tasks were announced at the conference, and these proceedings also include overview papers for the shared tasks, summarizing the results, as well as providing information about the data used and any procedures that were followed in conducting or scoring the tasks. In addition, there are short papers from each participating team that describe their underlying system in greater detail.

Like in previous years, we have received a far larger number of submissions than we could accept for presentation. WMT 2018 has received 84 full research paper submissions (not counting withdrawn submissions). This is a record number of research paper submissions and more than double the number of submissions of earlier editions of WMT. In total, WMT 2018 featured 27 full research paper presentations (32% acceptance rate) and 82 shared task poster presentations.

We would like to thank the members of the Program Committee for their timely reviews. We also would like to thank the participants of the shared task and all the other volunteers who helped with the evaluations.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, Karin Verspoor, and Mark Fishel

Co-Organizers

**Organizers**:

Ondřej Bojar (Charles University in Prague)
Rajen Chatterjee (FBK)
Christian Federmann (MSR)
Yvette Graham (DCU)
Barry Haddow (University of Edinburgh)
Matthias Huck (LMU Munich)
Antonio Jimeno Yepes (IBM Research Australia)
Philipp Koehn (Johns Hopkins University)
Christof Monz (University of Amsterdam)
Matteo Negri (FBK)
Aurélie Névéol (LIMSI, CNRS)
Mariana Neves (German Federal Institute for Risk Assessment)
Matt Post (Johns Hopkins University)
Lucia Specia (University of Sheffield)
Marco Turchi (FBK)
Karin Verspoor (University of Melbourne)
Mark Fishel (University of Tartu)

**Program Committee**:

Tamer Alkhouli (RWTH Aachen University)
Antonios Anastasopoulos (University of Notre Dame)
Tim Anderson (Air Force Research Laboratory)
Yuki Arase (Osaka University)
Mihael Arcan (INSIGHT, NUI Galway)
Duygu Ataman (Fondazione Bruno Kessler, University of Trento)
Eleftherios Avramidis (German Research Center for Artificial Intelligence (DFKI))
Amittai Axelrod (Amazon)
Parnia Bahar (RWTH Aachen University)
Ankur Bapna (Google)
Petra Barancikova (Charles University in Prague, Faculty of Mathematics and Physics)
Joost Bastings (University of Amsterdam)
Meriem Beloucif (University of Copenhagen)
Graeme Blackwood (IBM Research AI)
Frédéric Blain (University of Sheffield)
Chris Brockett (Microsoft Research)
Bill Byrne (University of Cambridge)
Ozan Caglayan (LIUM, Le Mans University)
Marine Carpuat (University of Maryland)

Francisco Casacuberta (Universitat Politècnica de València)
Sheila Castilho (Dublin City University)
Daniel Cer (Google Research)
Rajen Chatterjee (Fondazione Bruno Kessler)
Boxing Chen (Alibaba)
Colin Cherry (Google)
Mara Chinea-Rios (Universitat Politècnica de València)
Vishal Chowdhary (MSR)
Chenhui Chu (Osaka University)
Ann Clifton (Amazon)
Marta R. Costa-jussà (Universitat Politècnica de Catalunya)
Josep Crego (SYSTRAN)
James Cross (Facebook)
Raj Dabre (NICT)
Praveen Dakwale (University of Amsterdam)
Steve DeNeefe (SDL Research)
Michael Denkowski (Amazon.com, Inc.)
Mattia Antonino Di Gangi (Fondazione Bruno Kessler)
Miguel Domingo (PRHLT Center)
Kevin Duh (Johns Hopkins University)
Marc Dymetman (Naver Labs Europe)
Hiroshi Echizen'ya (Hokkai-Gakuen University)
Sergey Edunov (Faceook AI Research)
Micha Elsner (The Ohio State University)
Marcello Federico (FBK)
Yang Feng (Institute of Computing Technology, Chinese Academy of Sciences)
Andrew Finch (Apple Inc.)
Orhan Firat (Google AI)
Mark Fishel (University of Tartu)
Mikel L. Forcada (Universitat d'Alacant)
George Foster (Google)
Atsushi Fujita (National Institute of Information and Communications Technology)
Juri Ganitkevitch (Johns Hopkins University)
Mercedes García-Martínez (Pangeanic)
Ekaterina Garmash (KLM Royal Dutch Airlines)
Niyu Ge (IBM Research)
Ulrich Germann (University of Edinburgh)
Jesús González-Rubio (WebInterpret)
Isao Goto (NHK)
Cyril Goutte (National Research Council Canada)
Roman Grundkiewicz (School of Informatics, University of Edinburgh)
Mandy Guo (Google)
Thanh-Le Ha (Karlsruhe Institute of Technology)
Nizar Habash (New York University Abu Dhabi)

Gholamreza Haffari (Monash University)
Viktor Hangya (Ludwig-Maximilians-Universität München)
Greg Hanneman (Amazon)
Christian Hardmeier (Uppsala universitet)
Eva Hasler (SDL Research)
Yifan He (Alibaba Inc.)
John Henderson (MITRE)
Felix Hieber (Amazon Research)
Hieu Hoang (University of Edinburgh)
Vu Cong Duy Hoang (The University of Melbourne)
Ke Hu (ADAPT Research Centre, SALIS, Dublin City University)
Gonzalo Iglesias (SDL)
Kenji Imamura (National Institute of Information and Communications Technology)
Aizhan Imankulova (Tokyo Metropolitan University)
Julia Ive (University of Sheffield)
Marcin Junczys-Dowmunt (Microsoft)
Shahram Khadivi (eBay)
Huda Khayrallah (The Johns Hopkins University)
Yunsu Kim (RWTH Aachen University)
Rebecca Knowles (Johns Hopkins University)
Julia Kreutzer (Department of Computational Linguistics, Heidelberg University)
Roland Kuhn (National Research Council of Canada)
Shankar Kumar (Google)
Anoop Kunchukuttan (IIT Bombay)
Surafel Melaku Lakew (University of Trento and Fondazione Bruno Kessler)
Ekaterina Lapshinova-Koltunski (Universität des Saarlandes)
Alon Lavie (Carnegie Mellon University)
Gregor Leusch (eBay)
William Lewis (Microsoft Research)
Qun Liu (Huawei Noah's Ark Lab)
Samuel Läubli (University of Zurich)
Gideon Maillette de Buy Wenniger (ADAPT Centre - Dublin City University)
Andreas Maletti (Universität Leipzig)
Saab Mansour (Apple)
Krzysztof Marasek (Polish-Japanese Academy of Information Technology)
André F. T. Martins (Unbabel, Instituto de Telecomunicacoes)
Sameen Maruf (Monash University)
Rebecca Marvin (Johns Hopkins University)
Arne Mauser (Google, Inc)
Arya D. McCarthy (Johns Hopkins University)
Nikita Mediankin (Charles University in Prague)
Antonio Valerio Miceli Barone (The University of Edinburgh)
Paul Michel (Carnegie Mellon University)
Kenton Murray (University of Notre Dame)

Tomáš Musil (Charles University in Prague)
Mathias Müller (University of Zurich)
Masaaki Nagata (+81-774-93-5235)
Toshiaki Nakazawa (Japan Science and Technology Agency)
Preslav Nakov (Qatar Computing Research Institute, HBKU)
Graham Neubig (Carnegie Mellon University)
Jan Niehues (Karlsruhe Institute of Technology)
Xing Niu (University of Maryland)
Tsuyoshi Okita (Kyushuu institute of technology university)
Daniel Ortiz-Martínez (Technical University of Valencia)
Myle Ott (Facebook AI Research)
Carla Parra Escartín (ADAPT Centre / Dublin City University)
Pavel Pecina (Charles University)
Stephan Peitz (Apple)
Sergio Penkale (Lingo24)
Mārcis Pinnis (Tilde)
Martin Popel (Charles University, Faculty of Mathematics and Physics, UFAL)
Maja Popović (ADAPT Centre @ DCU)
Chris Quirk (Microsoft Research)
Preethi Raghavan (IBM Research TJ Watson)
Matīss Rikters (Tilde)
Annette Rios (Institute of Computational Linguistics, University of Zurich)
Devendra Sachan (CMU / Petuum Inc.)
Elizabeth Salesky (Carnegie Mellon University)
Hassan Sawaf (Amazon AWS)
Carolina Scarton (University of Sheffield)
Julian Schamper (RWTH Aachen University)
Helmut Schmid (CIS, Ludwig-Maximilians-Universität)
Jean Senellart (SYSTRAN)
Rico Sennrich (University of Edinburgh)
Patrick Simianer (Lilt Inc.)
Linfeng Song (University of Rochester)
Felix Stahlberg (University of Cambridge, Department of Engineering)
Dario Stojanovski (LMU Munich)
Brian Strope (Google)
Sara Stymne (Uppsala University)
Katsuhito Sudoh (Nara Institute of Science and Technology (NAIST))
Felipe Sánchez-Martínez (Universitat d'Alacant)
Aleš Tamchyna (Charles University in Prague, UFAL MFF)
Jörg Tiedemann (University of Helsinki)
Ke Tran (University of Amsterdam)
Yulia Tsvetkov (Carnegie Mellon University)
Marco Turchi (Fondazione Bruno Kessler)
Ferhan Ture (Comcast Applied AI Research)

Nicola Ueffing (eBay)
Masao Utiyama (NICT)
Eva Vanmassenhove (DCU)
Dušan Variš (Charles University, Institute of Formal and Applied Linguistics)
David Vilar (Amazon Research)
Martin Volk (University of Zurich)
Ekaterina Vylomova (PhD Student, University of Melbourne)
Wei Wang (Google Research)
Weiyue Wang (RWTH Aachen University)
Taro Watanabe (Google)
Marion Weller-Di Marco (University of Amsterdam)
Philip Williams (University of Edinburgh)
Hua Wu (Baidu)
Joern Wuebker (Lilt, Inc.)
Hainan Xu (Johns Hopkins University)
Yinfei Yang (Google)
François Yvon (LIMSI/CNRS)
Dakun Zhang (SYSTRAN)

# Table of Contents

# Conference Program

**Wednesday, October 31, 2018**

8:45–9:00 Opening Remarks

**9:00–10:30 Session 1: Shared Tasks Overview Presentations I**

9:00–9:30 *Findings of the 2018 Conference on Machine Translation (WMT18)*
Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn and Christof Monz

9:30–9:50 *Findings of the Third Shared Task on Multimodal Machine Translation*
Loïc Barrault, Fethi Bougares, Lucia Specia, Chiraag Lala, Desmond Elliott and Stella Frank

9:50–10:10 *Findings of the WMT 2018 Biomedical Translation Shared Task: Evaluation on Medline test sets*
Mariana Neves, Antonio Jimeno Yepes, Aurélie Névéol, Cristian Grozea, Amy Siu, Madeleine Kittner and Karin Verspoor

**Boaster Session for Research Papers Presented as Posters**

10:16–10:18 *Scaling Neural Machine Translation*
Myle Ott, Sergey Edunov, David Grangier and Michael Auli

10:18–10:20 *Character-level Chinese-English Translation through ASCII Encoding*
Nikola Nikolov, Yuhuang Hu, Mi Xue Tan and Richard H.R. Hahnloser

10:20–10:22 *Neural Machine Translation of Logographic Language Using Sub-character Level Information*
Longtu Zhang and Mamoru Komachi

10:22–10:24 *An Analysis of Attention Mechanisms: The Case of Word Sense Disambiguation in Neural Machine Translation*
Gongbo Tang, Rico Sennrich and Joakim Nivre

10:24–10:26 *Discourse-Related Language Contrasts in English-Croatian Human and Machine Translation*
Margita Šoštarić, Christian Hardmeier and Sara Stymne

10:26–10:28 *Coreference and Coherence in Neural Machine Translation: A Study Using Oracle Experiments*
Dario Stojanovski and Alexander Fraser

10:28–10:30 *A Large-Scale Test Set for the Evaluation of Context-Aware Pronoun Translation in Neural Machine Translation*
Mathias Müller, Annette Rios, Elena Voita and Rico Sennrich

**Wednesday, October 31, 2018 (continued)**

10:30-11:00 Coffee Break

**11:00–12:30 Session 2: Poster Session I**

*An Empirical Study of Machine Translation for the Shared Task of WMT18*
Chao Bei, Hao Zong, Yiming Wang, Baoyong Fan, Shiqi Li and Conghu Yuan

*Robust parfda Statistical Machine Translation Results*
Ergun Biçici

*The TALP-UPC Machine Translation Systems for WMT18 News Shared Translation Task*
Noe Casas, Carlos Escolano, Marta R. Costa-jussà and José A. R. Fonollosa

*Phrase-based Unsupervised Machine Translation with Compositional Phrase Embeddings*
Maksym Del, Andre Tättar and Mark Fishel

*Alibaba's Neural Machine Translation Systems for WMT18*
Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, Changfeng Zhu and Boxing Chen

*The RWTH Aachen University English-German and German-English Unsupervised Neural Machine Translation Systems for WMT 2018*
Miguel Graça, Yunsu Kim, Julian Schamper, Jiahui Geng and Hermann Ney

*Cognate-aware morphological segmentation for multilingual neural translation*
Stig-Arne Grönroos, Sami Virpioja and Mikko Kurimo

*The AFRL WMT18 Systems: Ensembling, Continuation and Combination*
Jeremy Gwinnup, Tim Anderson, Grant Erdmann and Katherine Young

*The University of Edinburgh's Submissions to the WMT18 News Translation Task*
Barry Haddow, Nikolay Bogoychev, Denis Emelin, Ulrich Germann, Roman Grundkiewicz, Kenneth Heafield, Antonio Valerio Miceli Barone and Rico Sennrich

*TencentFmRD Neural Machine Translation for WMT18*
Bojie Hu, Ambyer Han and Shen Huang

**Wednesday, October 31, 2018 (continued)**

*The MLLP-UPV German-English Machine Translation System for WMT18*
Javier Iranzo-Sánchez, Pau Baquero-Arnal, Gonçal V. Garcés Díaz-Munío, Adrià Martínez-Villaronga, Jorge Civera and Alfons Juan

*Microsoft's Submission to the WMT2018 News Translation Task: How I Learned to Stop Worrying and Love the Data*
Marcin Junczys-Dowmunt

*CUNI Submissions in WMT18*
Tom Kocmi, Roman Sudarikov and Ondřej Bojar

*The JHU Machine Translation Systems for WMT 2018*
Philipp Koehn, Kevin Duh and Brian Thompson

*JUCBNMT at WMT2018 News Translation Task: Character Based Neural Machine Translation of Finnish to English*
Sainik Kumar Mahata, Dipankar Das and Sivaji Bandyopadhyay

*NICT's Neural and Statistical Machine Translation Systems for the WMT18 News Translation Task*
Benjamin Marie, Rui Wang, Atsushi Fujita, Masao Utiyama and Eiichiro Sumita

*PROMT Systems for WMT 2018 Shared Translation Task*
Alexander Molchanov

*NTT's Neural Machine Translation Systems for WMT 2018*
Makoto Morishita, Jun Suzuki and Masaaki Nagata

*The Karlsruhe Institute of Technology Systems for the News Translation Task in WMT 2018*
Ngoc-Quan Pham, Jan Niehues and Alexander Waibel

*Tilde's Machine Translation Systems for WMT 2018*
Marcis Pinnis, Matiss Rikters and Rihards Krišlauks

*CUNI Transformer Neural MT System for WMT18*
Martin Popel

*The University of Helsinki submissions to the WMT18 news task*
Alessandro Raganato, Yves Scherrer, Tommi Nieminen, Arvi Hurskainen and Jörg Tiedemann

**Wednesday, October 31, 2018 (continued)**

*The RWTH Aachen University Supervised Machine Translation Systems for WMT 2018*
Julian Schamper, Jan Rosendahl, Parnia Bahar, Yunsu Kim, Arne Nix and Hermann Ney

*The University of Cambridge's Machine Translation Systems for WMT18*
Felix Stahlberg, Adrià de Gispert and Bill Byrne

*The LMU Munich Unsupervised Machine Translation Systems*
Dario Stojanovski, Viktor Hangya, Matthias Huck and Alexander Fraser

*Tencent Neural Machine Translation Systems for WMT18*
Mingxuan Wang, Li Gong, Wenhuan Zhu, Jun Xie and Chao Bian

*The NiuTrans Machine Translation System for WMT18*
Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao and Jingbo Zhu

*The University of Maryland's Chinese-English Neural Machine Translation Systems at WMT18*
Weijia Xu and Marine Carpuat

*EvalD Reference-Less Discourse Evaluation for WMT18*
Ondřej Bojar, Jiří Mírovský, Kateřina Rysová and Magdaléna Rysová

*The WMT'18 Morpheval test suites for English-Czech, English-German, English-Finnish and Turkish-English*
Franck Burlot, Yves Scherrer, Vinit Ravishankar, Ondřej Bojar, Stig-Arne Grönroos, Maarit Koponen, Tommi Nieminen and François Yvon

*Testsuite on Czech–English Grammatical Contrasts*
Silvie Cinkova and Ondřej Bojar

*A Pronoun Test Suite Evaluation of the English–German MT Systems at WMT 2018*
Liane Guillou, Christian Hardmeier, Ekaterina Lapshinova-Koltunski and Sharid Loáiciga

*Fine-grained evaluation of German-English Machine Translation based on a Test Suite*
Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt and Hans Uszkoreit

*The Word Sense Disambiguation Test Suite at WMT18*
Annette Rios, Mathias Müller and Rico Sennrich

**Wednesday, October 31, 2018 (continued)**

**Wednesday, October 31, 2018 (continued)**

*Neural Machine Translation with the Transformer and Multi-Source Romance Languages for the Biomedical WMT 2018 task*
Brian Tubay and Marta R. Costa-jussà

12:30–14:00 Lunch

**14:00–15:20 Session 3: Research Papers on Embedding and Context**

14:00–14:20 *Beyond Weight Tying: Learning Joint Input-Output Embeddings for Neural Machine Translation*
Nikolaos Pappas, Lesly Miculicich and James Henderson

14:20–14:40 *A neural interlingua for multilingual machine translation*
Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang and Jason Sun

14:40–15:00 *Improving Neural Language Models with Weight Norm Initialization and Regularization*
Christian Herold, Yingbo Gao and Hermann Ney

15:00–15:20 *Contextual Neural Model for Translating Bilingual Multi-Speaker Conversations*
Sameen Maruf, André F. T. Martins and Gholamreza Haffari

15:30–16:00 Coffee Break

**16:00–17:20 Session 4: Research Papers on Analysis and Data**

16:00–16:20 *Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation*
Antonio Toral, Sheila Castilho, Ke Hu and Andy Way

16:20–16:40 *Freezing Subnetworks to Analyze Domain Adaptation in Neural Machine Translation*
Brian Thompson, Huda Khayrallah, Antonios Anastasopoulos, Arya D. McCarthy, Kevin Duh, Rebecca Marvin, Paul McNamee, Jeremy Gwinnup, Tim Anderson and Philipp Koehn

16:40–17:00 *Denoising Neural Machine Translation Training with Trusted Data and Online Data Selection*
Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa and Ciprian Chelba

17:00–17:20 *Using Monolingual Data in Neural Machine Translation: a Systematic Study*
Franck Burlot and François Yvon

**Thursday, November 1, 2018**

**9:00–10:30 Session 5: Shared Tasks Overview Presentations I**

9:00–9:20 *Results of the WMT18 Metrics Shared Task: Both characters and embeddings achieve good performance*
Qingsong Ma, Ondřej Bojar and Yvette Graham

9:20–9:40 *Findings of the WMT 2018 Shared Task on Quality Estimation*
Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón Astudillo and André F. T. Martins

9:40–9:55 *Findings of the WMT 2018 Shared Task on Automatic Post-Editing*
Rajen Chatterjee, Matteo Negri, Raphael Rubino and Marco Turchi

9:55–10:10 *Findings of the WMT 2018 Shared Task on Parallel Corpus Filtering*
Philipp Koehn, Huda Khayrallah, Kenneth Heafield and Mikel L. Forcada

**Boaster Session for Research Papers Presented as Posters**

10:14–10:16 *Neural Machine Translation into Language Varieties*
Surafel Melaku Lakew, Aliia Erofeeva and Marcello Federico

10:16–10:18 *Effective Parallel Corpus Mining using Bilingual Sentence Embeddings*
Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-hsuan Sung, Brian Strope and Ray Kurzweil

10:18–10:20 *On The Alignment Problem In Multi-Head Attention-Based Neural Machine Translation*
Tamer Alkhouli, Gabriel Bretschner and Hermann Ney

10:20–10:22 *A Call for Clarity in Reporting BLEU Scores*
Matt Post

10:22–10:24 *Exploring gap filling as a cheaper alternative to reading comprehension questionnaires when evaluating machine translation for gisting*
Mikel L. Forcada, Carolina Scarton, Lucia Specia, Barry Haddow and Alexandra Birch

10:24–10:26 *Simple Fusion: Return of the Language Model*
Felix Stahlberg, James Cross and Veselin Stoyanov

10:26–10:28 *Correcting Length Bias in Neural Machine Translation*
Kenton Murray and David Chiang

10:28–10:30 *Extracting In-domain Training Corpora for Neural Machine Translation Using Data Selection Methods*
Catarina Cruz Silva, Chao-Hong Liu, Alberto Poncelas and Andy Way

10:30-11:00 Coffee Break

**11:00–12:30 Session 6: Poster Session II**

*Meteor++: Incorporating Copy Knowledge into Machine Translation Evaluation*
Yinuo Guo, Chong Ruan and Junfeng Hu

*ITER: Improving Translation Edit Rate through Optimizable Edit Costs*
Joybrata Panja and Sudip Kumar Naskar

*RUSE: Regressor Using Sentence Embeddings for Automatic Machine Translation Evaluation*
Hiroki Shimanaka, Tomoyuki Kajiwara and Mamoru Komachi

*Keep It or Not: Word Level Quality Estimation for Post-Editing*
Prasenjit Basu, Santanu Pal and Sudip Kumar Naskar

*RTM results for Predicting Translation Performance*
Ergun Biçici

*Neural Machine Translation for English-Tamil*
Himanshu Choudhary, Aditya Kumar Pathak, Rajiv Ratan Saha and Ponnurangam Kumaraguru

*The Benefit of Pseudo-Reference Translations in Quality Estimation of MT Output*
Melania Duma and Wolfgang Menzel

*Supervised and Unsupervised Minimalist Quality Estimators: Vicomtech's Participation in the WMT 2018 Quality Estimation Task*
Thierry Etchegoyhen, Eva Martínez Garcia and Andoni Azpeitia

**Thursday, November 1, 2018 (continued)**

*STACC, OOV Density and N-gram Saturation: Vicomtech's Participation in the WMT 2018 Shared Task on Parallel Corpus Filtering*
Andoni Azpeitia, Thierry Etchegoyhen and Eva Martínez garcia

*A hybrid pipeline of rules and machine learning to filter web-crawled parallel corpora*
Eduard Barbu and Verginica Barbu Mititelu

*Coverage and Cynicism: The AFRL Submission to the WMT 2018 Parallel Corpus Filtering Task*
Grant Erdmann and Jeremy Gwinnup

*MAJE Submission to the WMT2018 Shared Task on Parallel Corpus Filtering*
Marina Fomicheva and Jesús González-Rubio

*An Unsupervised System for Parallel Corpus Filtering*
Viktor Hangya and Alexander Fraser

*Dual Conditional Cross-Entropy Filtering of Noisy Parallel Corpora*
Marcin Junczys-Dowmunt

*The JHU Parallel Corpus Filtering Systems for WMT 2018*
Huda Khayrallah, Hainan Xu and Philipp Koehn

*Measuring sentence parallelism using Mahalanobis distances: The NRC unsupervised submissions to the WMT18 Parallel Corpus Filtering shared task*
Patrick Littell, Samuel Larkin, Darlene Stewart, Michel Simard, Cyril Goutte and Chi-kiu Lo

*Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The NRC supervised submissions to the Parallel Corpus Filtering task*
Chi-kiu Lo, Michel Simard, Darlene Stewart, Samuel Larkin, Cyril Goutte and Patrick Littell

*Alibaba Submission to the WMT18 Parallel Corpus Filtering Task*
Jun Lu, Xiaoyu Lv, Yangbin Shi and Boxing Chen

*UTFPR at WMT 2018: Minimalistic Supervised Corpora Filtering for Machine Translation*
Gustavo Paetzold

*The ILSP/ARC submission to the WMT 2018 Parallel Corpus Filtering Shared Task*
Vassilis Papavassiliou, Sokratis Sofianopoulos, Prokopis Prokopidis and Stelios Piperidis

**Thursday, November 1, 2018 (continued)**

*SYSTRAN Participation to the WMT2018 Shared Task on Parallel Corpus Filtering*
Minh Quang Pham, Josep Crego and Jean Senellart

*Tilde's Parallel Corpus Filtering Methods for WMT 2018*
Marcis Pinnis

*The RWTH Aachen University Filtering System for the WMT 2018 Parallel Corpus Filtering Task*
Nick Rossenbach, Jan Rosendahl, Yunsu Kim, Miguel Graça, Aman Gokrani and Hermann Ney

*Prompsit's submission to WMT 2018 Parallel Corpus Filtering shared task*
Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz Rojas and Gema Ramírez

*NICT's Corpus Filtering Systems for the WMT18 Parallel Corpus Filtering Task*
Rui Wang, Benjamin Marie, Masao Utiyama and Eiichiro Sumita

12:30–14:00 Lunch

**14:00–15:30 Session 3: Invited Talk**

15:30–16:00 Coffee Break

**16:00–17:20 Session 8: Research Papers on Multilingual Translation**

16:00–16:20 *Massively Parallel Cross-Lingual Learning in Low-Resource Target Language Translation*
Zhong Zhou, Matthias Sperber and Alexander Waibel

16:20–16:40 *Trivial Transfer Learning for Low-Resource Neural Machine Translation*
Tom Kocmi and Ondřej Bojar

16:40–17:00 *Input Combination Strategies for Multi-Source Transformer Decoder*
Jindřich Libovický, Jindřich Helcl and David Mareček

17:00–17:20 *Parameter Sharing Methods for Multilingual Self-Attentional Translation Models*
Devendra Sachan and Graham Neubig

**Thursday, November 1, 2018 (continued)**

# Scaling Neural Machine Translation

**Myle Ott**$^{\triangle}$ **Sergey Edunov**$^{\triangle}$ **David Grangier**$^{\triangledown *}$ **Michael Auli**$^{\triangle}$
$^{\triangle}$Facebook AI Research, Menlo Park & New York.
$^{\triangledown}$Google Brain, Mountain View.

## Abstract

Sequence to sequence learning models still require several days to reach state of the art performance on large benchmark datasets using a single machine. This paper shows that reduced precision and large batch training can speedup training by nearly 5x on a single 8-GPU machine with careful tuning and implementation.[1] On WMT'14 English-German translation, we match the accuracy of Vaswani et al. (2017) in under 5 hours when training on 8 GPUs and we obtain a new state of the art of 29.3 BLEU after training for 85 minutes on 128 GPUs. We further improve these results to 29.8 BLEU by training on the much larger Paracrawl dataset. On the WMT'14 English-French task, we obtain a state-of-the-art BLEU of 43.2 in 8.5 hours on 128 GPUs.

## 1 Introduction

Neural Machine Translation (NMT) has seen impressive progress in the recent years with the introduction of ever more efficient architectures (Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017). Similar sequence-to-sequence models are also applied to other natural language processing tasks, such as abstractive summarization (See et al., 2017; Paulus et al., 2018) and dialog (Sordoni et al., 2015; Serban et al., 2017; Dusek and Jurcícek, 2016).

Currently, training state-of-the-art models on large datasets is computationally intensive and can require several days on a machine with 8 high-end graphics processing units (GPUs). Scaling training to multiple machines enables faster experimental turn-around but also introduces new challenges: How do we maintain efficiency in a distributed setup when some batches process faster than others (i.e., in the presence of *stragglers*)? How do larger batch sizes affect optimization and generalization performance? While stragglers primarily affect multi-machine training, questions about the effectiveness of large batch training are relevant even for users of commodity hardware on a single machine, especially as such hardware continues to improve, enabling bigger models and batch sizes.

In this paper, we first explore approaches to improve training efficiency on a single machine. By training with reduced floating point precision we decrease training time by 65% with no effect on accuracy. Next, we assess the effect of dramatically increasing the batch size from 25k to over 400k tokens, a necessary condition for large scale parallelization with synchronous training. We implement this on a single machine by accumulating gradients from several batches before each update. We find that by training with large batches and by increasing the learning rate we can further reduce training time by 40% on a single machine. Finally, we parallelize training across 16 machines and find that we can reduce training time by an additional 90% compared to a single machine.

Our improvements enable training a Transformer model on the WMT'16 En-De dataset to the same accuracy as Vaswani et al. (2017) in just 32 minutes on 128 GPUs and in under 5 hours on 8 GPUs. This same model trained to full convergence achieves a new state of the art of 29.3 BLEU in 85 minutes. These scalability improvements additionally enable us to train models on much larger datasets. We show that we can reach 29.8 BLEU on the same test set in less than 10 hours when trained on a combined corpus of WMT and Paracrawl data containing ~150M sentence pairs (i.e., over 30x more training data). Similarly, on the WMT'14 En-Fr task we obtain a state of the art BLEU of 43.2 in 8.5 hours on 128 GPUs.

*Work done while at Facebook AI Research.

[1] Our implementation is available at: https://www.github.com/pytorch/fairseq

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 1–9
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64001

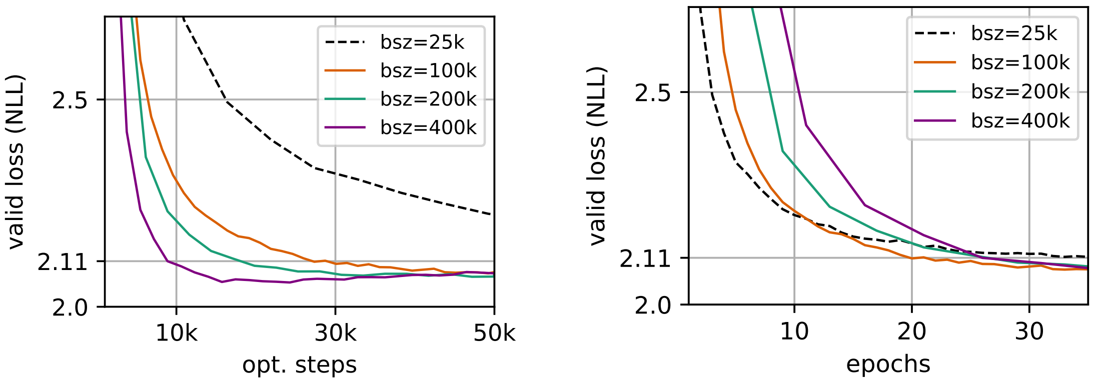


Figure 1: Validation loss for Transformer model trained with varying batch sizes (bsz) as a function of optimization steps (left) and epochs (right). Training with large batches is less data-efficient, but can be parallelized. Batch sizes given in number of target tokens excluding padding. *WMT En-De, newstest13.*

## 2 Related Work

Previous research considered training and inference with reduced numerical precision for neural networks (Simard and Graf, 1993; Courbariaux et al., 2015; Sa et al., 2018). Our work relies on half-precision floating point computation, following the guidelines of Micikevicius et al. (2018) to adjust the scale of the loss to avoid underflow or overflow errors in gradient computations.

Distributed training of neural networks follows two main strategies: (i) *model parallel* evaluates different model layers on different workers (Coates et al., 2013) and (ii) *data parallel* keeps a copy of the model on each worker but distributes different batches to different machines (Dean et al., 2012). We rely on the second scheme and follow synchronous SGD, which has recently been deemed more efficient than asynchronous SGD (Chen et al., 2016). Synchronous SGD distributes the computation of gradients over multiple machines and then performs a synchronized update of the model weights. Large neural machine translation systems have been recently trained with this algorithm with success (Dean, 2017; Chen et al., 2018).

Recent work by Puri et al. (2018) considers large-scale distributed training of language models (LM) achieving 109x scaling with 128 GPUs. Compared to NMT training, however, LM training does not face the same challenges of variable batch sizes. Moreover, we find that large batch training requires warming up the learning rate, whereas their work begins training with a large learning rate. There has also been recent work on using lower precision for inference only (Quinn and Ballesteros, 2018).

Another line of work explores strategies for improving communication efficiency in distributed synchronous training setting by abandoning “stragglers,” in particular by introducing redundancy in how the data is distributed across workers (Tandon et al., 2017; Ye and Abbe, 2018). The idea rests on coding schemes that introduce this redundancy and enable for some workers to simply not return an answer. In contrast, we do not discard any computation done by workers.

## 3 Experimental Setup

### 3.1 Datasets and Evaluation

We run experiments on two language pairs, English to German (En–De) and English to French (En–Fr). For En–De we replicate the setup of Vaswani et al. (2017) which relies on the WMT’16 training data with 4.5M sentence pairs; we validate on newstest13 and test on newstest14. We use a vocabulary of 32K symbols based on a joint source and target byte pair encoding (BPE; Sennrich et al. 2016). For En–Fr, we train on WMT’14 and borrow the setup of Gehring et al. (2017) with 36M training sentence pairs. We use newstest12+13 for validation and newstest14 for test. The 40K vocabulary is based on a joint source and target BPE factorization.

We also experiment with scaling training beyond 36M sentence pairs by using data from the Paracrawl corpus (ParaCrawl, 2018). This dataset is extremely large with more than 4.5B pairs for En–De and more than 4.2B pairs for

En–Fr. We rely on the BPE vocabulary built on WMT data for each language pair and explore filtering this noisy dataset in Section 4.5. We measure case-sensitive tokenized BLEU with `multi-bleu.pl`[2] and de-tokenized BLEU with SacreBLEU[3] (Post, 2018). All results use beam search with a beam width of 4 and length penalty of 0.6, following Vaswani et al. 2017. Checkpoint averaging is not used, except where specified otherwise.

### 3.2 Models and Hyperparameters

We use the Transformer model (Vaswani et al., 2017) implemented in PyTorch in the `fairseq-py` toolkit (Edunov et al., 2017). All experiments are based on the "big" transformer model with 6 blocks in the encoder and decoder networks. Each encoder block contains a self-attention layer, followed by two fully connected feed-forward layers with a ReLU non-linearity between them. Each decoder block contains self-attention, followed by encoder-decoder attention, followed by two fully connected feed-forward layers with a ReLU between them. We include residual connections (He et al., 2015) after each attention layer and after the combined feed-forward layers, and apply layer normalization (Ba et al., 2016) after each residual connection. We use word representations of size 1024, feed-forward layers with inner dimension 4,096, and multi-headed attention with 16 attention heads. We apply dropout (Srivastava et al., 2014) with probability 0.3 for En-De and 0.1 for En-Fr. In total this model has 210M parameters for the En-De dataset and 222M parameters for the En-Fr dataset.

Models are optimized with Adam (Kingma and Ba, 2015) using $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1\mathrm{e}{-8}$. We use the same learning rate schedule as Vaswani et al. (2017), i.e., the learning rate increases linearly for 4,000 steps to $5\mathrm{e}{-4}$ (or $1\mathrm{e}{-3}$ in experiments that specify `2x lr`), after which it is decayed proportionally to the inverse square root of the number of steps. We use label smoothing with 0.1 weight for the uniform prior distribution over the vocabulary (Szegedy et al., 2015; Pereyra et al., 2017).

All experiments are run on DGX-1 nodes with 8 NVIDIA© V100 GPUs interconnected by Infiniband. We use the NCCL2 library and `torch.distributed` for inter-GPU communication.

## 4 Experiments and Results

In this section we present results for improving training efficiency via reduced precision floating point (Section 4.1), training with larger batches (Section 4.2), and training with multiple nodes in a distributed setting (Section 4.3).

### 4.1 Half-Precision Training

NVIDIA Volta GPUs introduce Tensor Cores that enable efficient half precision floating point (FP) computations that are several times faster than full precision operations. However, half precision drastically reduces the range of floating point values that can be represented which can lead to numerical underflows and overflows (Micikevicius et al., 2018). This can be mitigated by scaling values to fit into the FP16 range.

In particular, we perform all forward-backward computations as well as the all-reduce (gradient synchronization) between workers in FP16. In contrast, the model weights are also available in full precision, and we compute the loss and optimization (e.g., momentum, weight updates) in FP32 as well. We scale the loss right after the forward pass to fit into the FP16 range and perform the backward pass as usual. After the all-reduce of the FP16 version of the gradients with respect to the weights we convert the gradients into FP32 and restore the original scale of the values before updating the weights.

In the beginning stages of training, the loss needs to be scaled down to avoid numerical overflow, while at the end of training, when the loss is small, we need to scale it up in order to avoid numerical underflow. Dynamic loss scaling takes care of both. It automatically scales down the loss when overflow is detected and since it is not possible to detect underflow, it scales the loss up if no overflows have been detected over the past 2,000 updates.

To evaluate training with lower precision, we first compare a baseline transformer model trained on 8 GPUs with 32-bit floating point (Our reimplementation) to the same model trained with 16-

[2] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`

[3] SacreBLEU hash: `BLEU+case.mixed+lang.en-{de,fr}+numrefs.1+smooth.exp+test.wmt14/full+tok.13a+version.1.2.9`

| model | # gpu | bsz | cumul | BLEU | updates | tkn/sec | time | speedup |
|---|---|---|---|---|---|---|---|---|
| Vaswani et al. (2017) | 8×P100 | 25k | 1 | 26.4 | 300k | ~25k | ~5,000 | – |
| Our reimplementation | 8×V100 | 25k | 1 | 26.4 | 192k | 54k | 1,429 | reference |
| + `16-bit` | 8 | 25k | 1 | 26.7 | 193k | 143k | 495 | 2.9x |
| + `cumul` | 8 | 402k | 16 | 26.7 | 13.7k | 195k | 447 | 3.2x |
| + `2x lr` | 8 | 402k | 16 | 26.5 | 9.6k | 196k | 311 | 4.6x |
| + `5k tkn/gpu` | 8 | 365k | 10 | 26.5 | 10.3k | 202k | 294 | 4.9x |
| 16 nodes (from +`2x lr`) | 128 | 402k | 1 | 26.5 | 9.5k | 1.53M | 37 | 38.6x |
| + `overlap comm+bwd` | 128 | 402k | 1 | 26.5 | 9.7k | 1.82M | 32 | 44.7x |

Table 1: Training time (min) for reduced precision (`16-bit`), cumulating gradients over multiple backwards (`cumul`), increasing learning rate (`2x lr`) and computing each forward/backward with more data due to memory savings (`5k tkn/gpu`). Average time (excl. validation and saving models) over 3 random seeds to reach validation perplexity of 4.32 (2.11 NLL). Cumul=16 means a weight update after accumulating gradients for 16 backward computations, simulating training on 16 nodes. *WMT En-De, newstest13.*

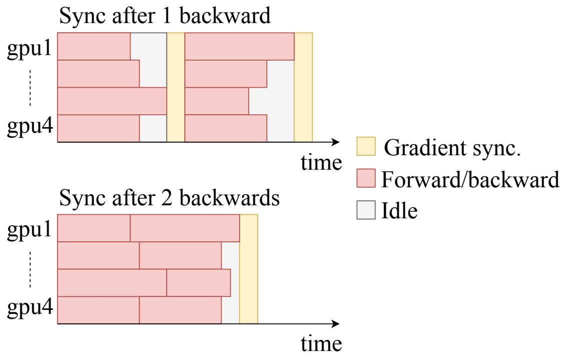


Figure 2: Accumulating gradients over multiple forward/backward steps speeds up training by: (i) reducing communication between workers, and (ii) saving idle time by reducing variance in workload between GPUs.

bit floating point (`16-bit`). Note, that we keep the batch size and other parameters equal. Table 1 reports training speed of various setups to reach validation perplexity 4.32 and shows that `16-bit` results in a 2.9x speedup.

### 4.2 Training with Larger Batches

Large batches are a prerequisite for distributed synchronous training, since it averages the gradients over all workers and thus the effective batch size is the sum of the sizes of all batches seen by the workers.

Figure 1 shows that bigger batches result in slower initial convergence when measured in terms of epochs (i.e. passes over the training set). However, when looking at the number of weight updates (i.e. optimization steps) large batches converge faster (Hoffer et al., 2017). These results support parallelization since the number of steps define the number of synchronization points for synchronous training.

Training with large batches is also possible on a single machine regardless of the number of GPUs or amount of available memory; one simply iterates over multiple batches and accumulates the resulting gradients before committing a weight update. This has the added benefit of reducing communication and reducing the variance in workload between different workers (see Figure 2), leading to a 36% increase in tokens/sec (Table 1, `cumul`). We discuss the issue of workload variance in more depth in Section 5.

**Increased Learning Rate**: Similar to Goyal et al. (2017) and Smith et al. (2018) we find that training with large batches enables us to increase the learning rate, which further shortens training time even on a single node (`2x lr`).

**Memory Efficiency**: Reduced precision also decreases memory consumption, allowing for larger sub-batches per GPU. We switch from a maximum of 3.5k tokens per GPU to a maximum of 5k tokens per GPU and obtain an additional 5% speedup (cf. Table 1; `2x lr` vs. `5k tkn/gpu`).

Table 1 reports our speed improvements due to reduced precision, larger batches, learning rate increase and increased per-worker batch size. Overall, we reduce training time from 1,429 min to 294 min to reach the same perplexity on the same hardware (8x NVIDIA V100), i.e. a 4.9x speedup.

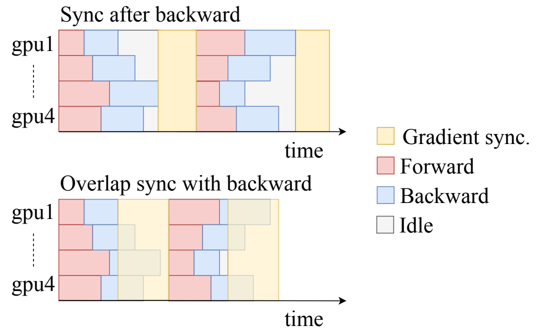


Figure 3: Illustration of how the backward pass in back-propagation can be overlapped with gradient synchronization to improve training speed.

### 4.3 Parallel Training

While large batch training improves training time even on a single node, another benefit of training with large batches is that it is easily parallelized across multiple nodes (machines). We run our previous 1-node experiment over 16 nodes of 8 GPUs each (NVIDIA V100), interconnected by Infiniband. Table 1 shows that with a simple, synchronous parallelization strategy over 16 nodes we can further reduce training time from 311 minutes to just 37 minutes (cf. Table 1; `2x lr` vs. `16 nodes`).

However, the time spent communicating gradients across workers increases dramatically when training with multiple nodes. In particular, our models contain over 200M parameters, therefore multi-node training requires transferring 400MB gradient buffers between machines. Fortunately, the sequential nature of back-propagation allows us to further improve multi-node training performance by beginning this communication in the background, while gradients are still being computed for the mini-batch (see Figure 3). Back-propagation proceeds sequentially from the top of the network down to the inputs. When the gradient computation for a layer finishes, we add the result to a synchronization buffer. As soon as the size of the buffer reaches a predefined threshold[4] we synchronize the buffered gradients in a background thread that runs concurrently with back-propagation down the rest of the network. Table 1 shows that by overlapping gradient communication with computation in the backwards pass, we can further reduce training time by 15%, from 37 minutes to just 32 minutes (cf. Table 1; `16 nodes` vs. `overlap comm+bwd`).

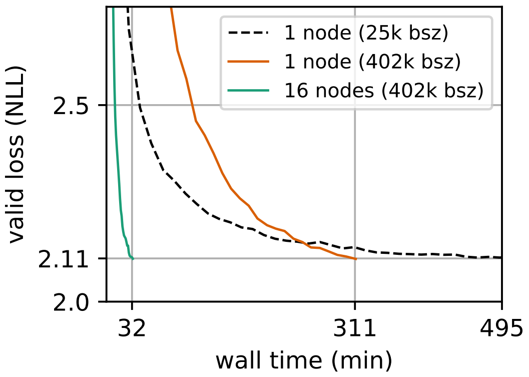


Figure 4: Validation loss (negative log likelihood on newstest13) versus training time on 1 vs 16 nodes.

| | En–De | En–Fr |
|---|---|---|
| a. Gehring et al. (2017) | 25.2 | 40.5 |
| b. Vaswani et al. (2017) | 28.4 | 41.0 |
| c. Ahmed et al. (2017) | 28.9 | 41.4 |
| d. Shaw et al. (2018) | 29.2 | 41.5 |
| Our result | **29.3** | **43.2** |
| *16-node training time* | *85 min* | *512 min* |

Table 2: BLEU on newstest2014 for WMT English-German (En–De) and English-French (En–Fr). All results are based on WMT'14 training data, except for En–De (b), (c), (d) and our result which are trained on WMT'16.

We illustrate the speedup achieved by large batches and parallel training in Figure 4.

### 4.4 Results with WMT Training Data

We report results on newstest14 for English-to-German (En-De) and English-to-French (En-Fr). For En-De, we train on the filtered version of WMT'16 from Vaswani et al. (2017). For En-Fr, we follow the setup of Gehring et al. (2017). In both cases, we train a "big" transformer on 16 nodes and average model parameters from the last 10 checkpoints (Vaswani et al., 2017). Table 2 reports 29.3 BLEU for En-De in 1h 25min and 43.2 BLEU for En-Fr in 8h 32min. We therefore establish a new state-of-the-art for both datasets, excluding settings with additional training data (Kutylowski, 2018). In contrast to Table 1, Table 2 reports times to convergence, not times to a specific validation likelihood.

[4] We use a threshold of 150MB in this work.

| Train set | En–De | En–Fr |
|---|---|---|
| WMT only | 29.3 | **43.2** |
| *detok. SacreBLEU* | *28.6* | *41.4* |
| *16-node training time* | *85 min* | *512 min* |
| WMT + Paracrawl | **29.8** | 42.1 |
| *detok. SacreBLEU* | *29.3* | *40.9* |
| *16-node training time* | *539 min* | *794 min* |

Table 3: Test BLEU (*newstest14*) when training with WMT+Paracrawl data.

## 4.5 Results with WMT & Paracrawl Training

Fast parallel training lets us additionally explore training over larger datasets. In this section we consider Paracrawl (ParaCrawl, 2018), a recent dataset of more than 4B parallel sentences for each language pair (En-De and En-Fr).

Previous work on Paracrawl considered training only on filtered subsets of less than 30M pairs (Xu and Koehn, 2017). We also filter Paracrawl by removing sentence-pairs with a source/target length ratio exceeding 1.5 and sentences with more than 250 words. We also remove pairs for which the source and target are copies (Ott et al., 2018). On En–De, this brings the set from 4.6B to 700M. We then train a En–De model on a clean dataset (WMT'14 news commentary) to score the remaining 700M sentence pairs, and retain the 140M pairs with best average token log-likelihood. To train an En–Fr model, we filter the data to 129M pairs using the same procedure.

Next, we explored different ways to weight the WMT and Paracrawl data. Figure 5 shows the validation loss for En-De models trained with different sampling ratios of WMT and filtered Paracrawl data during training. The model with 1:1 ratio performs best on the validation set, outperforming the model trained on only WMT data. For En-Fr, we found a sampling ratio of 3:1 (WMT:Paracrawl) performed best.

Test set results are given in Table 3. We find that Paracrawl improves BLEU on En–De to 29.8 but it is not beneficial for En–Fr, achieving just 42.1 vs. 43.2 BLEU for our baseline.

## 5 Analysis of Stragglers

In a distributed training setup with synchronized SGD, workers may take different amounts of time to compute gradients. Slower workers, or stragglers, cause other workers to wait. There are several reasons for stragglers but here we focus on the different amounts of time it takes to process the data on each GPU.

In particular, each GPU typically processes one *sub-batch* containing sentences of similar lengths, such that each sub-batch has at most $N$ tokens (e.g., $N = 3.5$k tokens), with padding added as required. We refer to sub-batches as the data that is processed on each GPU worker whose combination is the entire batch. The sub-batches processed by a worker may therefore differ from other workers in the following three characteristics: the number of sentences, the maximum source sentence length, or the maximum target sentence length. To illustrate how these characteristics impact training

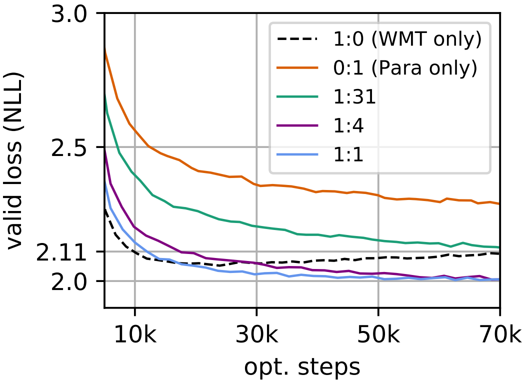


Figure 5: Validation loss when training on Paracrawl+WMT with varying sampling ratios. 1:4 means sampling 4 Paracrawl sentences for every WMT sentence. *WMT En-De, newstest13*.

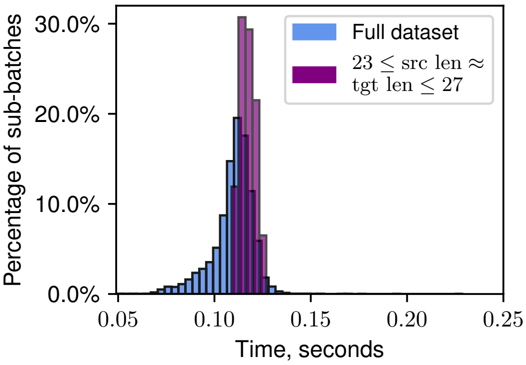


Figure 6: Histogram of time to complete one forward and backward pass for each sub-batch in the *WMT En-De* training dataset. Sub-batches consist of a variable number of sentences of similar length, such that each sub-batch contains at most 3.5k tokens.

speed, Figure 6 shows the amount of time required to process the 44K sub-batches in the En-De training data. There is large variability in the amount time to process sub-batches with different characteristics: the mean time to process a sub-batch is 0.11 seconds, the slowest sub-batch takes 0.228 seconds and the fastest 0.049 seconds. Notably, there is much less variability if we only consider batches of a similar shape (e.g., batches where $23 \leq \text{src len} \approx \text{tgt len} \leq 27$).

Unsurprisingly, constructing sub-batches based on a maximum token budget as just described exacerbates the impact of stragglers. In Section 4.2 we observed that we could reduce the variance between workers by accumulating the gradients over multiple sub-batches on each worker before updating the weights (see illustration in Figure 2). A more direct, but naïve solution is to assign all workers sub-batches with a similar shape. However, this increases the variance of the gradients across batches and adversely affects the final model. Indeed, when we trained a model in this way, then it failed to converge to the target validation perplexity of 4.32 (cf. Table 1).

As an alternative, we construct sub-batches so that each one takes approximately the same amount of processing time across all workers. We first set a target for the amount of time a sub-batch should take to process (e.g., the 90th percentile in Figure 6) which we keep fixed across training. Next, we build a table to estimate the processing time for a sub-batch based on the number of sentences and maximum source and target sentence lengths. Finally, we construct each worker's sub-batches by tuning the number of sentences until the estimated processing time reaches our target. This approach improves single-node throughput from 143k tokens-per-second to 150k tokens-per-second, reducing the training time to reach 4.32 perplexity from 495 to 479 minutes (cf. Table 1, `16-bit`). Unfortunately, this is less effective than training with large batches, by accumulating gradients from multiple sub-batches on each worker (cf. Table 1, `cumul`, 447 minutes). Moreover, large batches additionally enable increasing the learning rate, which further improves training time (cf. Table 1, `2x lr`, 311 minutes).

## 6 Conclusions

We explored how to train state-of-the-art NMT models on large scale parallel hardware. We investigated lower precision computation, very large batch sizes (up to 400k tokens), and larger learning rates. Our careful implementation speeds up the training of a big transformer model (Vaswani et al., 2017) by nearly 5x on one machine with 8 GPUs.

We improve the state-of-the-art for WMT'14 En-Fr to 43.2 vs. 41.5 for Shaw et al. (2018), training in less than 9 hours on 128 GPUs. On WMT'14 En-De test set, we report 29.3 BLEU vs. 29.2 for Shaw et al. (2018) on the same setup, training our model in 85 minutes on 128 GPUs. BLEU is further improved to 29.8 by scaling the training set with Paracrawl data.

Overall, our work shows that future hardware will enable training times for large NMT systems that are comparable to phrase-based systems (Koehn et al., 2007). We note that multi-node parallelization still incurs a significant overhead: 16-node training is only ∼10x faster than 1-node training. Future work may consider better batching and communication strategies.

## References

Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. 2017. Weighted transformer network for machine translation. *arxiv*, 1711.02132.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

Jianmin Chen, Rajat Monga, Samy Bengio, and Rafal Józefowicz. 2016. Revisiting distributed synchronous sgd. *Arxiv*, 1604.00981.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. *arxiv*, 1804.09849.

Adam Coates, Brody Huval, Tao Wang, David J. Wu, Bryan Catanzaro, and Andrew Y. Ng. 2013. Deep learning with cots hpc systems. In *Proc. of ICML*.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2015. Training deep neural networks with low precision multiplications.

Jeff Dean. 2017. Machine learning for systems and systems for machine learning. In *Proc. of NIPS Workshop on ML Systems*.

Jeffrey Dean, Gregory S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew W. Senior, Paul A. Tucker, Ke Yang, and Andrew Y. Ng. 2012. Large scale distributed deep networks. In *Proc. of NIPS*.

Ondrej Dusek and Filip Jurcícek. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proc. of ACL*.

Sergey Edunov, Myle Ott, and Sam Gross. 2017. Fairseq. `https://github.com/pytorch/fairseq`.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proc. of ICML*.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. In *Proc. of CVPR*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. In *Proc. of CVPR*.

Elad Hoffer, Itay Hubara, and Daniel Soudry. 2017. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Proc. of NIPS*, pages 1729–1739.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR*.

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. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL Demo Session*.

Jaroslaw Kutylowski. 2018. Deepl press information. `https://www.deepl.com/press.html`.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed Precision Training. In *Proc. of ICLR*.

Myle Ott, Michael Auli, David Grangier, and MarcAurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning (ICML)*.

ParaCrawl. 2018. ParaCrawl. `http://paracrawl.eu/download.html`.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *Proc. of ICLR*.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *Proc. of ICLR Workshop*.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv*, 1804.08771.

Raul Puri, Robert Kirby, Nikolai Yakovenko, and Bryan Catanzaro. 2018. Large scale language modeling: Converging on 40gb of text in four hours. *arXiv preprint arXiv:1808.01371*.

Jerry Quinn and Miguel Ballesteros. 2018. Pieces of eight: 8-bit neural machine translation. In *Proc. of NAACL*.

Christopher De Sa, Megan Leszczynski, Jian Zhang, Alana Marzoev, Christopher R. Aberger, Kunle Olukotun, and Christopher Ré. 2018. High-accuracy low-precision training. *Arxiv*, 1803.03383.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proc. of ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. of ACL*.

Iulian Serban, Alessandro Sordoni, Ryan Joseph Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proc. of AAAI*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proc. of NAACL*.

Patrice Y. Simard and Hans Peter Graf. 1993. Backpropagation without multiplication. In *Proc. of NIPS*.

Samuel L. Smith, Pieter-Jan Kindermans, and Quoc V. Le. 2018. Don't decay the learning rate, increase the batch size. In *Proc. of ICLR*.

Alessandro Sordoni, Michel Galley2, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proc. of ACL*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. *arXiv preprint arXiv:1512.00567*.

Rashish Tandon, Qi Lei, Alexandros G. Dimakis, and Nikos Karampatziakis. 2017. Gradient Coding: Avoiding Stragglers in Distributed Learning. In *Proc. of ICML*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proc. of NIPS*.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proc. of EMNLP*.

Min Ye and Emmanuel Abbe. 2018. Communication-Computation Efficient Gradient Coding. In *Proc. of ICML*.

# Character-level Chinese-English Translation through ASCII Encoding

**Nikola I. Nikolov***, **Yuhuang Hu***, **Mi Xue Tan**, **Richard H.R. Hahnloser**
Institute of Neuroinformatics, University of Zürich and ETH Zürich, Switzerland
{niniko, yuhuang.hu, mtan, rich}@ini.ethz.ch

## Abstract

Character-level Neural Machine Translation (NMT) models have recently achieved impressive results on many language pairs. They mainly do well for Indo-European language pairs, where the languages share the same writing system. However, for translating between Chinese and English, the gap between the two different writing systems poses a major challenge because of a lack of systematic correspondence between the individual linguistic units. In this paper, we enable character-level NMT for Chinese, by breaking down Chinese characters into linguistic units similar to that of Indo-European languages. We use the Wubi encoding scheme[1], which preserves the original shape and semantic information of the characters, while also being reversible. We show promising results from training Wubi-based models on the character- and subword-level with recurrent as well as convolutional models.

## 1 Introduction

Character-level sequence-to-sequence (Seq2Seq) models for machine translation can perform comparably to subword-to-subword or subword-to-character models, when dealing with Indo-European language pairs, such as German-English or Czech-English (Lee et al., 2017). Such language pairs benefit from having a common Latin character representation, which facilitates suitable character-to-character mappings to be learned. This method, however, is more difficult for non-Latin language pairs, such as Chinese-English. Chinese characters differ from English characters, in the sense that they carry more meaning and resemble subword units in English. For example, the Chinese character '人' corresponds to the word 'human' in English. This lack of correspondence makes the problem more demanding for a Chinese-English character-to-character model, as it would be forced to map higher-level linguistic units in Chinese to individual Latin characters in English. Good performance on this task may, therefore, require specific architectural decisions.

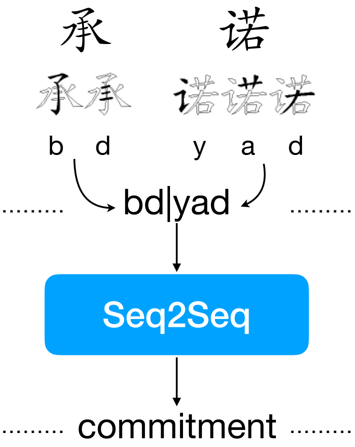


Figure 1: Overview of the **wubi2en** approach to Chinese-to-English translation. A raw Chinese word ('承诺') is encoded into ASCII characters ('bd|yad'), using the Wubi encoding method, before passing it to a Seq2Seq network. The network generates the English translation 'commitment', processing one ASCII character at a time.

In this paper, we propose a simple solution to this challenge: encode Chinese into a meaningful string of ASCII characters, using the **Wubi** method (Lunde, 2009) (Section 3). This encoding enables efficient and accurate character-level prediction applications in Chinese, with no changes required to the model architecture (see Figure 1). Our approach significantly reduces the character vocabulary size of a Chinese text, while preserving the shape and semantic information encoded in the Chinese characters.

[1] Code and data available at https://github.com/duguyue100/wmt-en2wubi.

* Equal contribution

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 10–16
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64002

We demonstrate the utility of the Wubi encoding on subword- and character-level Chinese NMT, comparing the performance of systems trained on Wubi vs. raw Chinese characters (Section 4). We test three types of Seq2Seq models: recurrent (Cho et al., 2014) convolutional (Gehring et al., 2017) as well as hybrid (Lee et al., 2017). Our results demonstrate the utility of Wubi as a preprocessing step for Chinese translation tasks, showing promising performance.

## 2 Background

### 2.1 Sequence-to-sequence models for NMT

Neural networks with Encoder-Decoder architectures have recently achieved impressive performance on many language pairs in Machine Translation, such as English-German and English-French (Wu et al., 2016). Recurrent Neural Networks (RNNs) (Cho et al., 2014) process and *encode* the input sequentially, mapping each word onto a vector representation of fixed dimensionality. The representations are used to condition a *decoder* RNN which generates the output sequence.

Recent studies have shown that Convolutional Neural Networks (CNNs) (LeCun et al., 1998) can perform better on Seq2Seq tasks than RNNs (Gehring et al., 2017; Chen and Wu, 2017; Kalchbrenner et al., 2016). CNNs enable simultaneous computations which are more efficient especially using parallel GPU hardware. Successive layers in CNN models have an increasing receptive field for modeling long-term dependencies in candidate languages.

### 2.2 Chinese-English translation

Recent large-scale benchmarks of RNN encoder-decoder models (Wu et al., 2016; Junczys-Dowmunt et al., 2016) have shown that translation pairs involving Chinese are among the most challenging for NMT systems. For instance, in Wu et al. (2016) an NMT system trained on English-to-Chinese had the least relative improvement across five other language pairs, measured over the performance of a phrase-based machine translation baseline.

While it is known that the quality of a Chinese translation system can be significantly impacted by the choice of word segmentation (Wang et al., 2015), there has been little work on improving the representation medium for Chinese translation. Wang et al. (2017) perform an empirical comparison on various translation granularities for the Chinese-English task. They find that adding additional information about the segmentation of the Chinese characters, such as marking the start and the end of each word, leads to improved performance over raw character or word translation.

The work that is most related to ours is (Du and Way, 2017), in which they use Pinyin[2] to romanize raw Chinese characters based on their pronunciation. This method, however, adds ambiguity to the data, because many Chinese characters share the same pronunciation.

## 3 Encoding Chinese characters with Wubi

**Wubi** (Lunde, 2009) is a shape-based encoding method for inputting Chinese characters on a computer QWERTY keyboard. The encoding is based on the structure of the characters rather than on their pronunciation. Using the method, each raw Chinese character (e.g., “设”) can be efficiently mapped to a unique sequence of 1 to 5 ASCII characters (e.g., “ymc”). This feature greatly reduces the ambiguity brought by other phonetic input methods, such as Pinyin.

As an input method, Wubi uses 25 key caps from the QWERTY keyboard, where each key cap is assigned to five categories based on the character’s first stroke (when written by hand). Each of the key caps is associated with different character roots. A Chinese character is broken down into its character roots, and a corresponding QWERTY association of the character roots is used to encode a word. For example, the Wubi encoding of ‘哈’ is ‘kwgk’, and the character roots of this word are 口(k), 人(w), 王(g) and 口(k). To create a one-to-one mapping of every Chinese character to a Wubi encoding during translation, we append numbers to the encodings, whenever one code maps to multiple Chinese characters.

Table 1: Examples of Wubi words and the corresponding Chinese words

| English | Chinese | Wubi |
|---|---|---|
| Set up | 编设 | xyna0\|ymc |
| Public property | 公共财产 | wc\|aw\|mf\|u |
| Step aside | 让开 | yh\|ga |

Applying Wubi significantly reduces the

[2] The official romanization system for Standard Chinese in mainland China.

character-level vocabulary size of a Chinese text (from $> 5,000$ commonly used Chinese characters, to 128 ASCII characters[3]), while preserving its shape and semantic information. Table 1 contains examples of Wubi, along with the corresponding words in Chinese and English.

## 4 Results

### 4.1 Dataset

In this work, we use a subset of the English and Chinese parts of the United Nations Parallel Corpus (Ziemski et al., 2016). We choose the UN corpus because of its high-quality, man-made translations. The dataset is sufficient for our purpose: our aim here is not to reach state-of-the-art performance on Chinese-English translation, but to demonstrate the potential of the Wubi encoding on the character level.

We preprocess the UN dataset with the MOSES tokenizer[4], and use Jieba[5] to segment the Chinese sentence into words, following which we encode the texts into Wubi. We use the '|' character as a subword separator for Wubi, in order to ensure that the mapping from Chinese to Wubi is unique. We also convert all Chinese punctuation marks (e.g. '。、《》') from UTF-8 to ASCII (e.g. '.,<>') because they share similar linguistic roles to English punctuations. This conversion additionally decreases the size of the Wubi character vocabulary.

Our final dataset contains 2.1M sentence pairs for training, and 55k pairs for validation and testing respectively (Table 2 contains additional statistics). Note that our procedures are entirely reversible.

Table 2: Statistics of our dataset (mean and standard deviation).

| | **English** | **Wubi** | **Chinese** |
|---|---|---|---|
| words per sentence | 25.8±11.0 | 22.9±10.0 | 22.9±10.0 |
| characters per word | 4.9±3.3 | 4.6±3.3 | 1.8±0.83 |
| characters per sentence | 152.3±67.9 | 127.1±56.5 | 63.5±27.6 |

To investigate the utility of the Wubi encoding, we compare the performance of NMT models on four training pairs: raw Chinese-to-English (*cn2en*) versus Wubi-to-English (*wubi2en*); English-to-raw Chinese (*en2cn*) versus English-to-Wubi (*en2wubi*). For each pair, we investigate three levels of sequence granularity: word-level, subword-level, and character-level. The word-level operates on individual English words (*e.g.* walk) and either raw-Chinese words (*e.g.* 编设) or Wubi words (*e.g.* sh|wy). We limit all word-level vocabularies to the 50k most frequent words for each language. The subword-level is produced using the byte pair encoding (BPE) scheme (Sennrich et al., 2016), capping the vocabulary size at 10k for each language. The character-level operates on individual raw-Chinese characters (e.g. '重'), or individual ASCII characters.

### 4.2 Model descriptions and training details

Our models are summarized in Table 3, including the number of parameters and vocabulary sizes used for each pair. For the subword- and word-level experiments, we use two systems[6]. The first, *LSTM*, is an LSTM Seq2Seq model (Cho et al., 2014) with an attention mechanism (Bahdanau et al., 2015). We use a single layer of 512 hidden units for the encoder and decoder, and set 512 as the embedding dimensionality. The second system, *FConv*, is a smaller version of the convolutional Seq2Seq model with an attention mechanism from (Gehring et al., 2017). We use word embeddings with dimension 256 for this model. The encoder and the decoder of *FConv* have the same convolutional architecture which consists of 4 convolution layers for the encoder and 3 for the decoder, each layer having filters with dimension 256 and size 3.

For all character-level experiments, we use the fully-character level model, *char2char* from (Lee et al., 2017)[7]. The encoder of this model consists of 8 convolutional layers with max pooling, which produce intermediate representations of segments of the input characters. Following this, a 4-layer highway network (Srivastava et al., 2015) is applied, as well as a single-layer recurrent network with gated recurrent units (GRUs) (Cho et al., 2014). The decoder consists of an attention mechanism and a two-layer GRU, which predicts the output one character at a time. The character embedding dimensionality is 128 for the encoder and

[3] 302 ASCII and special characters such as non-ASCII symbols used in the experiments, see Section 4.

[4] `https://github.com/moses-smt`

[5] `https://github.com/fxsjy/jieba`

[6] We use the `fairseq` library `https://github.com/pytorch/fairseq`.

[7] `https://github.com/nyu-dl/dl4mt-c2c`

Table 3: Model and vocabulary sizes used in our experiments. In brackets, we include the number of embedding parameters for a model (left), or the percentage of vocabulary coverage of the dataset (right).

| | No. of model parameters (Embedding) | | | Vocab Size (% coverage of dataset) | | |
|---|---|---|---|---|---|---|
| **level** | char2char | FConv | LSTM | EN | Wubi | CN |
| word | - | 42M (25M) | 83M (51M) | 50k (99.7%) | 50k (99.5%) | 50k (99.5%) |
| subword | - | 11M (5.1M) | 22M (10.6M) | 10k (100%) | 10k (100%) | 10k (98.7%) |
| character | 69-74M (0.21M-2.81M†) | - | - | 302 (100%) | 302 (100%) | 5183 (100%) |

†: 0.21M for wb2en/en2wb (69M in total); 0.77M for cn2en (70M) and 2.81M for en2cn (74M), due to a larger size of the decoder embedding.

Table 4: BLEU test scores on the UN dataset.

| | **character** | **subword** | | **word** | |
|---|---|---|---|---|---|
| | char2char | FConv | LSTM | FConv | LSTM |
| wubi2en | **40.55** | 38.20 | 43.06 | 39.53 | 43.36 |
| cn2en | 39.60 | 38.20 | 43.03 | 39.64 | 43.67 |
| en2wubi | **36.78** | **36.04** | **39.03** | 36.98 | 39.69 |
| en2cn† | 36.13 | 35.41 | 38.64 | 37.25 | 39.59 |

†: We convert these translations to Wubi before computing BLEU to ensure a consistent comparison.

512 for the decoder, whereas the number of hidden units is 512 for the encoder and 1024 for the decoder.

We train all models for 25 epochs using the Adam optimizer (Kingma and Ba, 2014). We used four NVIDIA Titan X GPUs for conducting the experiments, and use beam search with beam size of 20 to generate all final outputs.

### 4.3 Quantitative evaluation

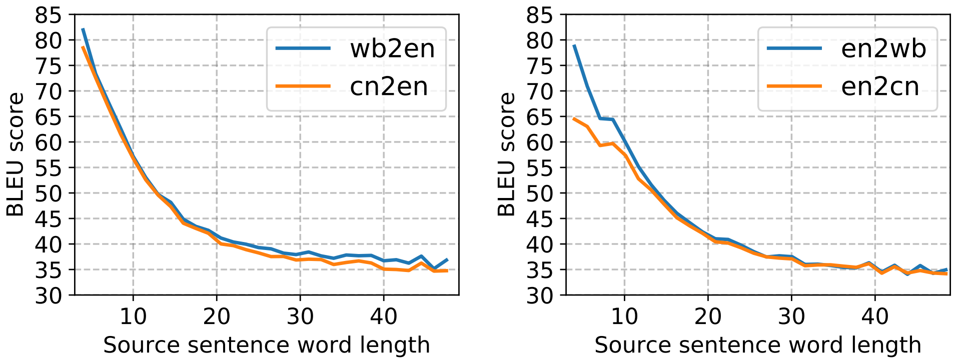
In Table 4, we present the BLEU scores for all the previously described experiments. Before computing BLEU, we convert all Chinese outputs to Wubi to ensure a consistent comparison. This conversion has a one-to-one mapping between Chinese and Wubi, whereas, in the reverse direction, ill-formed Wubi output on the character-level might not be reversible to Chinese.

On the word-level, the Wubi-based models achieve comparable results to their counterparts in Chinese, in both translation directions. *LSTM* significantly outperforms *FConv* across all experiments here, most likely due to its much larger size (see Table 3).

On the subword-level, we observe a slight increase of about 0.5 BLEU when translating from English to Wubi instead of raw Chinese. This increase is most likely due to the difference in the BPE vocabularies: while the English and Wubi BPE rules that were learned cover 100% of the dataset, for Chinese this is 98.7% - the remaining 1.3% had to be replaced by the *unk* symbol under our vocabulary constraints. While the models were capable of compensating for this gap when translating to English, in the reverse direction it resulted in a loss of performance. This highlights one benefit of Wubi on the subword-level: the Latin encoding seems to give a greater flexibility for extracting suitable BPE rules. It would be interesting to repeat this comparison using much larger datasets and larger BPE vocabularies.

Character-level translation is more difficult than word-level, since the models are expected to not only predict sentence-level semantics, but also to generate the correct spelling of each word. Our *char2char* Wubi models outperformed the raw Chinese models with 0.95 BLEU points when translating to English, and 0.65 BLEU when translating from English. The differences are statistically significant ($p = 0.001$ and $p = 0.034$ respectively) according to bootstrap resampling (Koehn, 2004) with 1500 samples. The results demonstrate the advantage of Wubi on the character-level, which outperforms raw Chinese even though it has fewer parameters dedicated for character embeddings (Table 3) and that it has to deal with substantially longer input or output sequences (see Table 2).

In Figure 2, we plot the sentence-level BLEU scores obtained by the *char2char* models on our test set, with respect to the length of the input sentences. When translating from Chinese to En-



(a) Translation from Chinese to English.

(b) Translation from English to Chinese.

Figure 2: Sentence-level BLEU scores obtained by the character-level *char2char* models on our test dataset, plotted with respect to the word length of the source sentences.

glish (Figure 2a) the Wubi-based model consistently outperforms the raw Chinese model, for all input lengths. Interestingly, the gap between the two systems increases for longer Chinese inputs of over 20 words, indicating that Wubi is more robust for such examples. This result could be explained by the fact that the encoder of the *char2char* model is more suitable for modeling languages with a higher level of granularity such as English and German. When translating from English to Chinese (Figure 2b) Wubi still has a small edge, however in this case we see the reverse trend: it performs much better on shorter sentences up to 12 English words. Perhaps, the increased granularity of the output sequence led to an advantage during decoding using beam search.

Interestingly, all the *char2char* models use only a tiny fraction of their parameters as embeddings, due to the much smaller size of their vocabularies. The best-performing LSTM word-level model has the majority of its parameters, 61% or over 50M, dedicated to word embeddings. For the Wubi-based character-level models, the number is only 0.3% or 0.21M. There is even a significant difference between Wubi and Chinese on the character-level, for example, *en2wb* has 12 times fewer embedding parameters than *en2cn*. Thus, although *char2char* performed worse than *LSTM* in our experiments, these results highlight the potential of character-level prediction for developing compact yet performant translation systems, for Latin as well as non-Latin languages.

### 4.4 Qualitative evaluation

In Table 5, we present four examples from our test dataset that cover short as well as long sentences. We also include the translations produced by the character-level *char2char* systems, which is the main focus of this paper. Full examples from the additional systems are available in the supplementary material.

In the first example, which is a short sentence resembling the headline of a document, both the *wubi2en* and cn2en models produced correct translations. When translating from English to Chinese, however, the *en2wubi* produced the word ‘与’ (highlighted in red) which more correctly matches the ground truth text. In contrast, the *en2cn* model produced the synonym ‘和’. In the second example, the *en2wubi* output completely matches the ground truth and is superior to the *en2cn* output. The latter failed to correctly translate ‘the’ to ‘这次’ (marked in green).

The *wubi2en* translation in the third example accurately translated the word ‘believe’ (marked in blue) and the full form of the abbreviation ‘ldcs’ – ‘the least developed countries’ (highlighted in green), whereas the *cn2en* chooses ‘are convinced’ and ignores ‘ldcs’ in its output sentence. Interestingly, although the ground truth text maps the word ‘essential’ (marked in red) to three Chinese words ‘至␣为␣重要’, both *en2wubi* and *en2cn* use only a single word to interpret it. Arguably, *en2wubi*’s translation ‘至关重要’ is closer to the ground truth than *en2cn*’s translation ‘必不可少’.

The fourth example is more challenging. There, the English ground truth ‘requested’ (highlighted in blue) maps to two different parts of the Chinese ground truth ‘提出’ (in blue) and ‘要求’ (in green). This one-to-many mapping confuses both translation models. The *wubi2en* tries to match the Chinese text by translating ‘提出’ into ‘pro-

Table 5: Four examples from our test dataset, along with system-generated translations produced by the *char2char* models. We converted the Wubi translations to raw Chinese. Translations of words with a similar meaning are marked with the same color.

| **Translation Type** | | **Example 1** |
|---|---|---|
| **English** | ground truth | social and human rights questions |
| **Chinese** | ground truth | 社会 与 人权 问题 |
| **Wubi** | ground truth | py\|wf gn w\|sc ukd0\|jghm1\| |
| | wubi2en | social and human rights questions |
| | cn2en | social and human rights questions |
| | en2wubi | 社会 与 人权 问题 |
| | en2cn | 社会 和 人权 问题 |
| | | **Example 2** |
| **English** | ground truth | the informal consultations is open to all member states . |
| **Chinese** | ground truth | 所有 会员国 均 可 参加 这次 非正式 协商 。 |
| **Wubi** | ground truth | rn\|e wf\|km\|l fqu sk cd\|lk p\|uqw djd\|ghd0\|aa fl\|um . |
| | wubi2en | this informal consultation may be open to all member states . |
| | cn2en | the informal consultations will be open to all member states . |
| | en2wubi | 所有 会员国 均 可 参加 这次 非正式 协商 。 |
| | en2cn | 所有 会员国 均 可 进行 非正式 协商 。 |
| | | **Example 3** |
| **English** | ground truth | we believe that increased trade is essential for the growth and development of ldcs . |
| **Chinese** | ground truth | 我们 相信 ， 增加 贸易 对 最 不 发达国家 的 增长 和 发展 至 为 重要 。 |
| **Wubi** | ground truth | q\|wu sh\|wy , fu\|lk qyv\|jqr cf jb i v\|dp\|l\|pe r fu\|ta t v\|nae gcf o tgj\|s . |
| | wubi2en | we believe that increased trade is essential for the growth and development of the least developed countries . |
| | cn2en | we are convinced that increased trade growth and development is essential . |
| | en2wubi | 我们 认为 ， 增加 贸易 对 最 不 发达国家 的 增长 和 发展 至关重要 。 |
| | en2cn | 我们 认为 ， 增加 贸易 对于 最 不 发达国家 的 增长 和 发展 来说 是 必不可少 的 。 |
| | | **Example 4** |
| **English** | ground truth | in some cases , additional posts were requested without explanation . |
| **Chinese** | ground truth | 在 某些 情况 中 ， 提出 增加 员额 要求 时 ， 并未 作出 说明 。 |
| **Wubi** | ground truth | d afs\|hxf nge\|ukq k , rj\|bm fu\|lk km\|ptkm0 s\|fiy jf , ua\|fii wt\|bm yu\|je . |
| | wubi2en | in some cases , no indication was made when additional staffing requirements were proposed . |
| | cn2en | in some cases , there was no indication of the request for additional posts . |
| | en2wubi | 在 有些 情况 下 ， 要求 增加 员额 。 |
| | en2cn | 在 有些 情况 下 还 要求 增设 员额 ， 但 没有 作出 任何 解释 。 |

posed' and '要求' into 'requirements': this model may have been misled by the word '时' (can be translated to 'when'); the output contains an adverbial clause. While the *wubi2en* output is closer to the ground truth, the two have little overlap. For the English-to-Chinese task, the *en2cn* translation is better than the one produced by *en2wubi*: while *en2cn* successfully translated 'without explanation' (in red), the *en2wubi* model ignored this part of the sentence.

The Wubi-based models tend to produce slightly shorter translations for both directions (see Table 6). In overall, the Wubi-based outputs appear to be visibly better than the raw Chinese-based outputs, in both directions.

## 5 Conclusion

We demonstrated that an intermediate encoding step to ASCII characters is suitable for the character-level Chinese-English translation task, and can even lead to performance improvements. All of our models trained using the Wubi encoding achieve comparable or better performance to the baselines trained directly on raw Chinese. On the character-level, using Wubi yields BLEU improvements when translating both to and from English, despite the increased length of the input or output sequences, and the smaller number of embedding parameters used. Furthermore, there are also improvements on the subword-level, when translating from English.

Table 6: Word counts of the outputs of the *char2char* models (mean and standard deviation).

| **Model** | **Word Count** |
|---|---|
| **wb2en** | 25.01 ± 10.95 |
| **cn2en** | 25.80 ± 11.72 |
| **en2wb** | 21.61 ± 9.68 |
| **en2cn** | 22.19 ± 10.11 |

Future work will focus on making use of the semantic structure of the Wubi encoding scheme, to develop architectures tailored to utilize it. Another exciting future direction is multilingual many-to-one character-level translation from Chinese and several Latin languages simultaneously, which becomes possible using encodings such as Wubi. This has previously been successfully realized for Latin and Cyrillic languages (Lee et al., 2017).

## Acknowledgments

We acknowledge support from the Swiss National Science Foundation (grant 31003A_156976) and from the National Centre of Competence in Research (NCCR) Robotics.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations 2015*.

Qiming Chen and Ren Wu. 2017. CNN is all you need. *CoRR*, abs/1712.09662.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. 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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734.

Jinhua Du and Andy Way. 2017. Pinyin as subword unit for chinese-sourced neural machine translation. In *Irish Conference on Artificial Intelligence and Cognitive Science 2017*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment? a case study on 30 translation directions. In *Proceedings of the International Workshop on Spoken Language Translation 2016*, volume 1.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aäron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *CoRR*, abs/1610.10099.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

K. Lunde. 2009. *CJKV Information Processing*. O'Reilly Series. O'Reilly Media.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2377–2385. Curran Associates, Inc.

Rui Wang, Hai Zhao, and Bao-Liang Lu. 2015. English to Chinese translation: How chinese character matters. In *PACLIC*.

Yining Wang, Long Zhou, Jiajun Zhang, and Chengqing Zong. 2017. Word, subword or character? an empirical study of granularity in chinese-english nmt. In *Machine Translation*, pages 30–42, Singapore. Springer Singapore.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *LREC*.

# Neural Machine Translation of Logographic Languages Using Sub-character Level Information

**Longtu Zhang**
Tokyo Metropolitan University
6-6 Asahigaoka, Hino,
Tokyo 191-0065, Japan
zhang-longtu@ed.tmu.ac.jp

**Mamoru Komachi**
Tokyo Metropolitan University
6-6 Asahigaoka, Hino,
Tokyo 191-0065, Japan
komachi@tmu.ac.jp

## Abstract

Recent neural machine translation (NMT) systems have been greatly improved by encoder-decoder models with attention mechanisms and sub-word units. However, important differences between languages with logographic and alphabetic writing systems have long been overlooked. This study focuses on these differences and uses a simple approach to improve the performance of NMT systems utilizing decomposed sub-character level information for logographic languages. Our results indicate that our approach not only improves the translation capabilities of NMT systems between Chinese and English, but also further improves NMT systems between Chinese and Japanese, because it utilizes the shared information brought by similar sub-character units.

## 1 Introduction

Neural machine translation (Cho et al., 2014) (NMT) systems based on sequence-to-sequence models (Sutskever et al., 2014) have recently become the de facto standard architecture. The models use attention mechanisms (Bahdanau et al., 2015; Luong et al., 2015) to keep records of all encoding results, and can focus on particular parts of these results during decoding, so that the model can produce longer and more accurate translations. Sub-word units are another technique first introduced by Sennrich's (2016) application of the byte pair encoding (BPE) algorithm, and are used to break up words in both source and target sentences into sequences of smaller units, learned without supervision. This alleviates the risk of producing <unk> symbols when the model encounters infrequent "unknown" words, also known as the out-of-vocabulary (OOV) problem. Moreover, sub-word units, which can be viewed as learned stems and affixes, can help the NMT model better encode the source sentence and decode the target sentence, particularly when the source and target languages share some similarities.

Almost all of the methods used to improve NMT systems were developed for alphabetic languages such as English, French, and German as either the source or target language, or both. An alphabetic language typically uses an alphabet: a small set of letters (basic writing symbols) that each roughly represents a phoneme in the spoken language. Words are composed by ordered letters, and sentences are composed by space-segmented ordered words. However, in other major writing systems—namely, logographic (or character-based) languages such as Chinese, Japanese, and traditional Korean—strokes are used to construct ideographs; ideographs are used to construct characters, which are the basic units for meaningful words. Words can then further compose sentences. In alphabetic languages, sub-word units are easy to identify, whereas in logographic languages, a similar effect can be achieved only if sub-character level information is taken into consideration.[1]

Having noticed this significant difference between these two writing systems, Shi et al. (2015), Liu et al. (2017), Peng et al. (2017), and Cao et al. (2017) used stroke-level information for logographic languages when constructing word embeddings; Toyama et al. (2017) used visual information for strokes and Japanese Kanji

[1] Taking the ASPEC corpus as an example, the average word lengths are roughly 1.5 characters (Chinese words, tokenized by Jieba tokenizer), 1.7 characters (Japanese words, tokenized by MeCab tokenizer), and 5.7 characters (English words, tokenized by Moses tokenizer), respectively. Therefore, when a sub-word model of similar vocabulary size is applied directly, English sub-words usually contain several letters, which are more effective in facilitating NMT, whereas Chinese and Japanese sub-words are largely just characters.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 17–25
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64003

radicals in a text classification task.[2]

Some studies have performed NMT tasks using various sub-word "equivalents". For instance, Du and Way (2017) trained factored NMT models using "Pinyin"[3] sequences on the source side. Unfortunately, they did not apply a BPE algorithm during training, and their model also cannot perform factored decoding. Wang et al. (2017) directly applied a BPE algorithm to character sequences before building NMT models. However, they did not take advantage of sub-character level information during the training of sub-word and NMT models. Kuang and Han (2018) also attempted to use a factored encoder for Chinese NMT systems using radical data. It is worth noting that although the idea of using ideographs and strokes in NLP tasks (particularly in NMT tasks) is not new, no previous NMT research has focused on the decoding process. If it is also possible to construct an ideograph/stroke decoder, we can further investigate translations between logographic languages. Additionally, no NMT research has previously used stroke data.

To summarize, there are three potential information gaps associated with current studies on NMT systems for logographic languages using sub-character level data: 1) no research has been performed on the decoding process; 2) no studies have trained models using sub-character level sub-words; and 3) no studies have attempted to build NMT models for logographic language pairs, despite their sharing many similarities. This study investigates whether sub-character information can facilitate both encoding and decoding in NMT systems and between logographic language pairs, and aims to determine the best sub-character unit granularity for each setting.

The main contributions of this study are threefold:

1. We create a sub-character database of Chinese character-based languages, and conduct MT experiments using various types of sub-character NMT models.
2. We facilitate the encoding or decoding process by using sub-character sequences on either the source or target side of the NMT system. This will improve translation performance; if sub-character information is shared between the encoder and decoder, it will further benefit the NMT system.
3. Specifically, Chinese ideograph[4] data and Japanese stroke data are the best choices for relevant NMT tasks.

## 2 Background

### 2.1 NMT with Attention Mechanisms and Sub-word Units

In this study, we applied a sequence-to-sequence model with an attention mechanism (Bahdanau et al., 2015). The basic recurrent unit is the "long short-term memory" (Hochreiter and Schmidhuber, 1997) unit. Because of the nature of the sequence-to-sequence model, the vocabulary size must be limited for the computational efficiency of the Softmax function. In such cases, the decoder outputs an `<unk>` symbol for any word that is not in the vocabulary, which will harm the translation quality. This is called the out-of-vocabulary (OOV) problem.

Sub-word unit algorithms (such as BPE algorithms) first break up a sentence into the smallest possible units. Then, two adjacent units at a time are merged according to some standard (e.g., the co-occurrence frequency). Finally, after $n$ steps, the algorithm collects the merged units as "sub-word" units. By using sub-word units, it is possible to represent a large number of words with a small vocabulary. Originally, sub-word units were only applied to unknown words (Sennrich et al., 2016). However, in the recent GNMT (Wu et al., 2016) and transformer systems (Vaswani et al., 2017), all words are broken up into sub-word units to better represent the shared information.

For alphabetic languages, researchers have indicated that sub-word units are useful for solving OOV problems, and that shared information can further improve translation quality. The Sentencepiece project[5] compared several combinations of word-pieces (Kudo, 2018) and BPE sub-word

[2] To be more precise, there is another so-called syllabic writing system, which uses individual symbols to represent symbols rather than phonemes. Japanese hiragana and katakana are actually syllabic symbols rather than ideographs. In this paper, we focus only on the logographic part.

[3] An official Romanization system for standard Chinese in mainland China. Pinyin includes both letters and diacritics, which represent phonemic and tonal information, respectively.

[4] We use the term "logographic" to refer to writing systems such as Chinese characters and Japanese Kanji, and "ideograph" to refer to the character components.

[5] https://github.com/google/sentencepiece

| Character | Semantic ideograph | Phonetic ideograph | Pinyin |
|---|---|---|---|
| 驰 run | 马 horse | 也 | chн |
| 池 pool | 水( 氵) water | 也 | chн |
| 施 impose | 方 direction | 也 | sh |
| 弛 loosen | 弓 bow | 也 | chн |
| 地 land | 土 soil | 也 | dм |
| 驱 drive | 马 horse | 区 | q |

Table 1: Examples of decomposed ideographs of Chinese characters. The composing ideographs of different functionality might be shared across different characters.

models in English/Japanese NMT tasks. The subword units were trained on character (Japanese Kanji and Hiragana/Katakana) sequences. Similarly, Wang et al. (2017) attempted to compare the effects of different segmentation methods on NMT tasks, including "BPE" units trained on Chinese character sequences.

### 2.2 Sub-character Units in NLP

In alphabetic languages, the smallest unit for sub-word unit training is the letter; in character-based languages, the smallest units should be sub-character units, such as ideographs or strokes. Because sub-character units are shared across different characters and have similar meanings, it is possible to build a significantly smaller vocabulary to cover a large amount of training data. This has been researched quite extensively within tasks such as word embeddings, as mentioned previously.

As we can see from the examples in Table 1, there are several independent Chinese characters. Each character can be split into at least two ideographs: a semantic ideograph and a phonetic ideograph.[6] More importantly, the same ideograph can be shared by different characters denoting similar meanings. For example, the first five characters (驰, 池, 施, 弛 and 地) have similar pronunciation (and they are written similarly in Pinyin) because they share the same phonetic ideograph "也". Similarly, semantic ideographs can be shared across characters and denote a similar semantic meaning. For example, the first character "驰" and the last character "驱" share same semantic ideograph "马" (meaning "horse"); and their semantic meanings are closely related ("run" and "drive", respectively). A few ideographs can also be treated as standalone characters.

| Word | Meaning | Ideographs |
|---|---|---|
| 树木 | Wood | 木对木 |
| 森林 | Forest | 木木木木木 |

Table 2: Examples of multi-character words in Chinese and their ideograph sequences.

To the best of our knowledge, however, no research has been performed on logographic language NMT beyond character-level data, except in the work of Du and Way (2017), who attempted to use Pinyin sequences instead of character sequences in Chinese–English NMT tasks. Considering the fact that there are a large number of homophones and homonyms in Chinese languages, it was difficult for this method to be used to reconstruct characters in the decoding step.

## 3 NMT Using Sub-character Level Units

### 3.1 Ideograph Information

When building NMT vocabulary, the use of sub-characters (instead of words, characters, and character level sub-words) can greatly condense vocabulary size. For example, a vocabulary can be decreased from 6,000 to 10,000 character types[7] to hundreds [8] of ideographs. Table 2 presents two Chinese words composed of four different characters that have very close meanings. Character-based NMT models treat these characters separately as one-hot vectors. In contrast, if the two words are broken down into ideograph sequences, they overlap significantly. Then, only two ideographs are needed to compose the vocabulary of the two words. The computational load will be reduced, and the chances of training neurons responsible for low-frequency vocabularies will increase.

Moreover, sub-character units can serve as building blocks for constructing characters that are not present in the training data, because all CJK characters are designed to be composed of a limited number of ideographs in UNICODE standards.

### 3.2 Stroke Information

All ideographs can be further decomposed into strokes, which are smaller units and have an even

[6] Semantic ideographs denote the meaning of a character, whereas phonetic ideographs denote the pronunciation.

[7] According to the ASPEC corpus.

[8] 214 as defined in UNICODE 10.0 standard and 517 as defined in CNS11643 charset.

smaller number of types. Therefore, we also propose training our model on stroke sequences. There are five basic stroke types for Chinese characters and Japanese Kanji: "horizontal" (一), "vertical" (丨), "right falling" (㇏), "left falling" (丿), and "break" (㇕). Each stroke type can be further sub-categorized into several stroke variations. For example, left falling strokes contain both long and short left fallings (丿 and ㇒), while a break contains many more variations, such as ㇄, ㇈, ㇆, and ㇉ (details can be found in Appendix A).

In practice, the CNS11643 charset[9] is used to transform each character into a stroke sequence, where unfortunately only "stroke-type" information is available. In this study, we manually transcribed all ideographs into stroke sequences using 33 pre-defined strokes.

### 3.3 Character Decomposition

The CNS11643 charset is used to facilitate character decomposition, where Chinese, Japanese, and Korean characters are merged into a single character type based on similarities in their forms and meanings. This is potentially beneficial; for example, if Chinese and Japanese vocabularies are built, they will authentically share some common types. There are 517 so-called "components" (i.e., ideographs) pre-defined in CNS11643. This ensures that all characters can be divided into certain sequences of components. For example, the character "可" can be split into "丁" and "口"; and the character "君" can be split into "尹" and "口". Details can be found on the CNS11643 website[10]. Using this ideograph decomposition information, all Chinese and Japanese sentence data can be transformed into new ideograph sequences; then, using the manually transcribed stroke decomposition data introduced in Section 3.2, we can also obtain new stroke sequences.

Note that although there are no clear indications of how the components/strokes are structured together, the sequence potentially contains structural information, because the writing of characters always follows a certain order, such as "up-down", "outside-in", etc. We also note that UNICODE 10.0 has introduced symbols indicating sub-character structures (Ideographic Description Characters), which provide a clearer indication of character compositions. We will make further use of this information in future studies.

| Language | Word |
|---|---|
| JP-character | 風 景 |
| JP-ideograph | 几一虫 日亠口小_1 |
| JP-stroke | 丿㇈一丨㇕一丨㇒丶<br>丨㇕一一丶一丨㇕一亅㇒丶_1 |
| CN-character | 风 景 |
| CN-ideograph | 几㐅 日亠口小_1 |
| CN-stroke | 丿㇈㇒丶<br>丨㇕一一丶一丨㇕一亅㇒丶_1 |
| EN | landscape |

Table 3: The Japanese word 風景 and Chinese word 风景 both mean "landscape" in English, and they only differ in the middle part of the first character. Note that there are "_1" tags at the ends of some decomposed sequences to distinguish between possible duplications.

To ensure that there are no duplicated ideograph and stroke sequences for different characters and multi-character words, we post-tag the sequences on the duplicated ones using "_1", "_2", etc. Table 3 shows an example of character decomposition in Chinese and Japanese[11].

## 4 Experiments on Chinese–Japanese–English Translation

To answer our research questions, we set up a series of experiments to compare NMT models of logographic languages trained on word sequences, character-level sub-word unit sequences, and ideograph- and stroke-level subword unit sequences.

We performed two lines of experiments:

1. We trained NMT models between logographic language and alphabetic language combinations, i.e., Japanese/Chinese and English. In each model, we varied the data granularity for the logographic language, using "character level" or "sub-character level" (ideograph level and stroke level) granularities. We used the character level

[9] The CNS11643 charset is published and maintained by the Taiwan government.
http://www.cns11643.gov.tw/AIDB/welcome_en.do

[10] http://www.cns11643.gov.tw/search.jsp?ID=13

[11] For example, the ideograph and stroke sequences for character 景 are the same as those for character 晾 (meaning "to dry in the sun"). However, these two characters have different architectures ("top-down" vs. "left-right"). Post-tags are thus appended in order to distinguish them. Similarly, characters 风 and 凬 have the same ideograph and stroke sequences, and thus must be post-tagged.

NMT models as our baselines, and investigated whether the sub-character level NMT models could outperform the baseline models.

2. We trained NMT models between combinations of two logographic languages, i.e., Chinese and Japanese. Similarly, we used data sets with different granularities: 1) Models lacking sub-character level data. 2) Models having sub-character level data on both sides (to confirm the results of the previous experiment). For the experiments, the models will have both source and target sides. The models will use sub-character level data with/without shared vocabularies (namely, ideograph models, stroke models, ideograph-stroke models, stroke-ideograph models, and ideograph/stroke models with shared vocabularies). 3) Pinyin baselines according to (Du and Way, 2017), where both Pinyin word sequences with tones and character sequences with Pinyin factors are used with the encoder.

### 4.1 Dataset

We trained our baselines and experiments using Chinese, Japanese, and English. The Asian Scientific Paper Excerpt Corpus (ASPEC (Nakazawa et al., 2016)) and Casia2015[12] corpus were used for this purpose.

ASPEC contains a Japanese–English paper abstract corpus of 3 million parallel sentences (ASPEC-JE) and a Japanese–Chinese paper excerpt corpus of 680,000 parallel sentences (ASPEC-JC). We used the first million confidently aligned parallel sentences in ASPEC-JE and all of the ASPEC-JC data to cover Japanese–English and Japanese–Chinese language pairs. The Casia2015 corpus contains approximately 1 million parallel Chinese–English sentences. All data in the Casia2015 corpus were used to cover Chinese–English language pairs. During training, the maximum length hyperparameter was adjusted to ensure 90% coverage of the training data. For development and testing, the ASPEC corpus has an official split between the development set and test set; however, because the Casia2015 corpus is not similarly split, we made random selections from the development set and test set of 1,000 sentences each.

### 4.2 Settings

Different pre-tokenization methods were applied to the data in three languages (if applicable). A Moses tokenizer was applied to the English data; a Jieba[13] tokenizer using the default dictionary was applied to the Chinese data; and a MeCab[14] tokenizer using the IPA dictionary was applied to the Japanese data. For the Pinyin baseline, the pypinyin[15] Python library was used to transcribe the Chinese character sequence into a Pinyin sequence.

In both of the experiment lines discussed above, data at the "word", "character", "ideograph", and "stroke" levels were used in combinations. For "word" level data, only dictionary-based segmentation was applied; for the other three levels of data, the byte pair encoding (BPE) models were trained and applied, with a vocabulary size of 8,000. In the second line of experiments, where both the source and target sides were logographic languages, we added "character" level data without BPE ("char") for comparison. Additionally, shared vocabularies were applied when both the source and target had the same data granularity level (meaning that both the source and target side would have the same vocabulary)[16].

A basic RNNsearch model (Bahdanau et al., 2015) with two layers of long short-term memory (LSTM) units was used. The hidden size was 512. A normalized Bahdanau attention mechanism was applied at the output layer of the decoder. We developed our model based on TensorFlow[17] and its neural machine translation tutorial[18].

The model was trained on a single GeForce GTX TITAN X GPU. During training, the SGD optimizer was used, and the learning rate was set at 1.0. The size of the training batch was set to 128, and the total global training step was 250,000. We also decayed the learning rate as the training progressed: after two-thirds of the train-

[12] http://nlp.nju.edu.cn/cwmt-wmt/, provided by the Institute of Automation, Chinese Academy of Sciences.

[13] https://github.com/fxsjy/jieba

[14] http://taku910.github.io/mecab/

[15] https://github.com/mozillazg/python-pinyin

[16] The shared vocabulary can be trained by a BPE model on a concatenated corpus of source and target sentences.

[17] https://github.com/tensorflow

[18] https://github.com/tensorflow/nmt

| **English-Japanese NMT** | | **BLEU** |
|---|---|---|
| EN_word | JP_word | 36.1 |
| EN_word | JP_character | 38.3 |
| EN_word | JP_ideograph | 40.3$^*$ |
| EN_word | JP_stroke | **41.3**$^*$ |
| **Japanese-English NMT** | | **BLEU** |
| JP_word | EN_word | 25.5 |
| JP_character | EN_word | 26.3 |
| JP_ideograph | EN_word | 26.8$^*$ |
| JP_stroke | EN_word | **27.0**$^*$ |
| **English-Chinese NMT** | | **BLEU** |
| EN_word | CN_word | 11.8 |
| EN_word | CN_character | 10.3 |
| EN_word | CN_ideograph | **14.6**$^*$ |
| EN_word | CN_stroke | 14.1$^*$ |
| **Chinese-English NMT** | | **BLEU** |
| CN_word | EN_word | 14.7 |
| CN_character | EN_word | 14.5 |
| CN_ideograph | EN_word | **15.6**$^*$ |
| CN_stroke | EN_word | 15.5$^*$ |

Table 4: Experimental results (BLEU scores) of NMT systems for Japanese/English and Chinese/English language pairs. All the scores are statistically significant at $p = 0.0001$ (marked by $*$).

ing steps, we set the learning rate to be four times smaller until the end of training. Additionally, we set the drop-out rate to 0.2 during training.

BLEU was used as the evaluation metric in our experiments. For Chinese and Japanese data, a KyTea tokenization was applied before we applied BLEU, following the WAT (Workshop on Asian Translation) leaderboard standard. To validate the significance of our results, we ran bootstrap re-sampling (Koehn, 2004) for all results using Travatar (Neubig, 2013) at a significance level of $p = 0.0001$.

## 4.3 Results

### 4.3.1 NMT of Logographic and Alphabetic Language Pairs

Table 4 shows the experimental results for the Japanese/English and Chinese/English language pairs in both translation directions. Generally, for each of the experiment settings, the models using ideograph and stroke data outperformed the baseline systems, regardless of the language pair or translation direction. However, for the Japanese/English language pair, the stroke sequence models performed better. For the Chinese/English language pairs, the ideograph sequence models worked better. The reason for these differences will be discussed in detail in Section 5.

### 4.3.2 NMT of Logographic Language Pairs

Table 5 shows the results for all baselines and proposed models. Among the character-level baselines, the “char” models and “bpe” models outperformed the “word” models in both translation directions. When we applied a shared vocabulary to the “bpe” models, the models achieved the best BLEU scores in both translation directions. These character-level baselines conform with previous studies indicating that sub-word units improve the performance of NMT systems, and that whenever both the source and target side data have similarities in their writing systems, shared vocabularies will further enhance performance.

Sub-character level models aim to replicate similar results to those presented in Section 4.3.1, because only one side of these models uses sub-character level data. For Japanese–Chinese translation directions, half of the models showed a significant improvement over the baselines, whereas for Chinese–Japanese translation directions, five out of six models showed significant improvements.

When both the source and target side used the same sub-character level data (either ideograph or stroke data), the experimental results also showed significant improvement over character baselines. Additionally, the ideograph models outperformed stroke models. When shared vocabularies were applied to the models, the ideograph models exhibited slight performance improvements ($0.1 \sim 0.4$ BLEU point), and the stroke models exhibited dramatically decreased performance ($0.9 \sim 1.1$ BLEU points). However, no model here outperformed the sub-character baselines.

To further exploit the power of sub-character units, the last models having different levels of sub-character units on the source and target side were trained. The results conform with what we found in Section 4.3.1: the models using Chinese ideograph data and Japanese stroke data exhibited the best performance, regardless of whether they were applied at the source or target side. For Japanese–Chinese translations, the best BLEU score was 33.8, which was produced by the Japanese-stroke and Chinese-ideograph model; for Chinese–Japanese translation, the best BLEU score was 43.9, which was produced by the Chinese-ideograph and Japanese-stroke model.

| JP-CN NMT | CN_word | CN_char | CN_bpe | CN_ideograph | CN_stroke |
|---|---|---|---|---|---|
| **JP_word** | 29.6 | - | - | 30.8 | 30.3 |
| **JP_char** | - | 31.6 | - | 32.0$^*$ | 32.1$^*$ |
| **JP_bpe** | - | - | 31.5 (31.7) | 31.6 | 31.7 |
| **JP_ideograph** | 30.4 | 33.1$^*$ | 33.3$^*$ | 32.0$^*$ (32.4$^*$) | 33.4$^*$ |
| **JP_stroke** | 30.3 | 33.4$^*$ | 32.6$^*$ | **33.8**$^*$ | 32.1$^*$ (31.2) |
| **CN-JP NMT** | **JP_word** | **JP_char** | **JP_bpe** | **JP_ideograph** | **JP_stroke** |
| **CN_word** | 40.0 (*40.0*) | - | - | 40.5 | 40.1 |
| **CN_char** | 42.1 (*40.4*) | 41.7 | - | 43.1$^*$ | 42.2$^*$ |
| **CN_bpe** | 42.1 | - | 42.0 (42.3) | 43.1$^*$ | 42.2$^*$ |
| **CN_ideograph** | 43.2$^*$ | 43.5$^*$ | 43.0$^*$ | 42.6$^*$ (42.7$^*$) | **43.9**$^*$ |
| **CN_stroke** | 43.0$^*$ | 43.3$^*$ | 42.5$^*$ | 42.9$^*$ | 42.2$^*$ (41.1) |

Table 5: Experimental results (BLEU scores) for Japanese/Chinese NMT systems. The row headers and column headers indicate which source and target data were used in the training. In particular, "word" and "char" are character level data without BPE segmentation, while "bpe" (character level), "ideograph", and "stroke" (sub-character level) are data with BPE segmentation. The scores in parentheses indicate the models that had a shared vocabulary, whenever applicable. The italic numbers represent the two Pinyin baselines used for comparative purposes, namely the "WdPyT" model, which uses Pinyin words with tones as the source data, and the "factored-NMT" model, which uses Pinyin characters as factors (Du and Way, 2017). Note that these two baselines can only have Chinese data on the encoder side. The $*$ superscripts indicate that a score is significantly better than the best baseline result.

## 5 Discussions

### 5.1 Translation Examples

Table 6 shows some of the translation examples. There is a rare proper noun "松下電器 (Matsushita Electric)" (OOV) in the source sentence. The word baseline model cannot decode this; therefore, an <unk> symbol is produced. The character baseline model avoids the OOV problem. However, the underlined parts in both baseline translations seem to be word-for-word translations from the Japanese source sentence ("松下 電器 グループ で は"), which become a prepositional phrase in Chinese ("在 松下 电器 集团 中 (in Matsushita Electric Group)"). This makes the translation ungrammatical because there will be no noun phrase as the subject in the sentence. Our best model (i.e., sub-character based NMT model using Japanese stroke data and Chinese ideograph data) can solve these two problems by better encoding the source sentence and can produce translations both without OOV and with a noun phrase as the sentence subject.

### 5.2 Strokes vs. Ideographs

The experimental results show that in NMT models, different logographic languages appear to prefer sub-character units with different granularities. A very clear tendency that was observed consistently in both experiments was that ideographs worked better for Chinese and strokes worked better for Japanese. This difference might be because of the differences in the writing systems. In addition to Kanji (Chinese characters), Japanese uses Hiragana and Katakana, which are standalone alphabets.

Moreover, as described in Section 4, stroke models tended to perform more poorly than ideograph models. This probably occurred because to achieve a fair comparison between all baseline models and proposed models, the same hyper-parameter configurations were used. For example, the embedding dimensions for all vocabularies were set to 300. This might be appropriate for vocabularies of character-based data and ideograph data having vocabulary sizes larger than 500. However, the stroke data only has a vocabulary size of approximately 30, which is too disproportional. This phenomenon might also account for the decrease in BLEU scores when shared vocabularies were applied to stroke models.

### 5.3 The Encoding and Decoding Process

In comparison with character level data, sub-character level data (such as ideographs and strokes) can be used to generate much smaller and more concentrated vocabularies. This is helpful during both the encoding and decoding processes. Vocabularies constructed using character-level data are known to be very skewed, containing both very frequent words and very rare

| Model | Sentence |
|---|---|
| Source | 松下 電器 グループ で は ，経営 理念 の 基本 と し て 1991 年 に 「 環境 宣言 」 を 制定 し た 。 |
| Reference | 作为 经营 理念 ，松下电气集团 于 1991年 制定 了 《 环境 宣言 》 。 |
| Baseline (Word) | 在 <unk> 集团 中 ，1991年 制定 了“ 环境 宣言 ”作为 经营 理念 的 基础 。 |
| Baseline (Char) | 在 松下电器 集团 中 ，作为 经营 理念 的 基础 ，1991年 制定 了 《 环境 宣言 》 。 |
| Best Model (JP-stroke-CN-ideograph) | 松下电器集团 ，作为 经营 理念 的 基础 ，1991年 制定 了“ 环境 宣言 ”。 |
| English Translation | The Matsushita Electric Group enacted the ”Environmental Declaration” as the basis of its business philosophy in 1991. |

Table 6: Translation examples of Japanese-Chinese NMT systems. Note that “松下电器” as a proper noun, could be handled properly in sub-character based translation systems.

words. As a result, during training, the neurons responsible for high-frequency words might be updated many times, while the neurons responsible for low-frequency words might be updated only a very limited number of times. This will potentially harm translation performance for low-frequency words.

However, this problem can be alleviated by applying sub-character units. Because ideographs and strokes are repeatedly shared by different characters, no items occur with very low frequencies. More instances can be found in the training data, even for the least frequent sub-character items. Therefore, the translation performance for low-frequency items could be much better.

## 6 Conclusions and Future Work

This study was the first attempt to use sub-character units in NMT models. Our results not only confirmed the positive effects of using ideograph and stroke sequences in NMT tasks, but also indicated that different logographic languages actually preferred different sub-character granularities (namely, ideograph for Chinese and stroke for Japanese). Finally, this paper presented a simple method for extending the available corpus from the character level to the sub-character level. During this process, we maintained a one-to-one relationship between the original characters and transformed sub-character sequences. As a result, this simple and straightforward method achieved consistently better results for NMT systems used to translate logographic languages, and could be easily applied to similar scenarios.

Many questions remain to be answered in future work. The first question involves the relative benefits of ideograph data and stroke data, and the effects of shared vocabularies. We have not yet explained why there are considerable differences in performance. In particular, for NMT models in which both sides have stroke data, why does performance drop dramatically when shared vocabularies are applied? We discussed the possible reasons for this phenomenon in Section 5.2; however, further investigation is needed.

Another important issue is as follows: when characters are transformed into ideographs and strokes, no structural information is considered. This causes repetitions in data, and we must add tags at the end of each sequence to differentiate them. A better way to solve this problem would be to have structural information directly encoded in the data.

## Acknowledgments

This work was partially supported by JSPS Grant-in-Aid for Young Scientists (B) Grant Number JP16K16117.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Shaosheng Cao, Wei Lu, Jun Zhou, and Xiaolong Li. 2017. Investigating stroke-level information for learning Chinese word embeddings. In *ISWC-16*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP 2014*, pages 1724–1734.

Jinhua Du and Andy Way. 2017. Pinyin as subword unit for Chinese-sourced neural machine translation. In *Irish Conference on Artificial Intelligence and Cognitive Science*, page 11.

Sepp Hochreiter and Jьrgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP 2014*.

Shaohui Kuang and Lifeng Han. 2018. Apply Chinese radicals into neural machine translation: Deeper than character level. In *30Th European Summer School In Logic, Language And Information*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL 2018*.

Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig. 2017. Learning character-level compositionality with visual features. In *ACL 2017*, pages 2059–2068.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP 2015*, pages 1412–1421.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In *LREC-9*, pages 2204–2208.

Graham Neubig. 2013. Travatar: A forest-to-string machine translation engine based on tree transducers. In *ACL 2013: System Demonstrations*, pages 91–96.

Haiyun Peng, Erik Cambria, and Xiaomei Zou. 2017. Radical-based hierarchical embeddings for Chinese sentiment analysis at sentence level. In *FLAIRS-30*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL 2016*, pages 1715–1725.

Xinlei Shi, Junjie Zhai, Xudong Yang, Zehua Xie, and Chao Liu. 2015. Radical embedding: Delving deeper to Chinese radicals. In *ACL-IJCNLP 2015*, pages 594–598.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

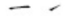
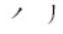
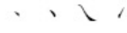
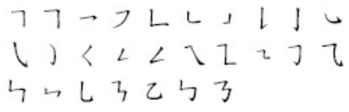
Yota Toyama, Makoto Miwa, and Yutaka Sasaki. 2017. Utilizing visual forms of Japanese characters for neural review classification. In *IJCNLP 2017*, pages 378–382.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS 2017*, pages 6000–6010.

Yining Wang, Long Zhou, Jiajun Zhang, and Chengqing Zong. 2017. Word, subword or character? an empirical study of granularity in Chinese-English NMT. In *China Workshop on Machine Translation*, pages 30–42. Springer.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144 [cs]*.

## Appendix A Strokes in CNS11643 Charset

| Type | Strokes |
|---|---|
| Horizontal |  |
| Vertical | |
| Left-falling |  |
| Right-falling |  |
| Break |  |

# An Analysis of Attention Mechanisms: The Case of Word Sense Disambiguation in Neural Machine Translation

Gongbo Tang[1] Rico Sennrich[2,3] Joakim Nivre[1]
[1]Department of Linguistics and Philology, Uppsala University
[2]School of Informatics, University of Edinburgh
[3]Institute of Computational Linguistics, University of Zurich
firstname.lastname@{lingfil.uu.se, ed.ac.uk}

## Abstract

Recent work has shown that the encoder-decoder attention mechanisms in neural machine translation (NMT) are different from the word alignment in statistical machine translation. In this paper, we focus on analyzing encoder-decoder attention mechanisms, in the case of word sense disambiguation (WSD) in NMT models. We hypothesize that attention mechanisms pay more attention to context tokens when translating ambiguous words. We explore the attention distribution patterns when translating ambiguous nouns. Counter-intuitively, we find that attention mechanisms are likely to distribute more attention to the ambiguous noun itself rather than context tokens, in comparison to other nouns. We conclude that attention is not the main mechanism used by NMT models to incorporate contextual information for WSD. The experimental results suggest that NMT models learn to encode contextual information necessary for WSD in the encoder hidden states. For the attention mechanism in Transformer models, we reveal that the first few layers gradually learn to "align" source and target tokens and the last few layers learn to extract features from the related but unaligned context tokens.

## 1 Introduction

Human languages exhibit many different types of ambiguity. Lexical ambiguity refers to the fact that words can have more than one semantic meaning. Dealing with these lexical ambiguities is a challenge for various NLP tasks. Word sense disambiguation (WSD) is recognizing the correct meaning of an ambiguous word, with the help of contextual information.

In statistical machine translation (SMT) (Koehn et al., 2003), a system could explicitly take context tokens into account to improve the translation of ambiguous words (Vickrey et al., 2005). By contrast, in neural machine translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014), especially in attentional NMT (Bahdanau et al., 2015; Luong et al., 2015), each hidden state incorporates contextual information. Hence, NMT models could potentially perform WSD well. However, there are no empirical results to indicate that the hidden states encode the contextual information needed for disambiguation. Moreover, how the attention mechanism[1] deals with ambiguous words is also not known yet.

In this paper, we focus on the question of how encoder-decoder attention mechanisms deal with ambiguous nouns. We explore two different attention mechanisms. One is the vanilla one-layer attention mechanism (Bahdanau et al., 2015; Luong et al., 2015), and the other one is the Transformer attention mechanism (Vaswani et al., 2017).

Rios et al. (2017) find that attentional NMT models perform well in translating ambiguous words with frequent senses,[2] while Liu et al. (2018) show that there are plenty of incorrect translations of ambiguous words. In Section 4, we evaluate the translations of ambiguous nouns, using the test set from Rios et al. (2017). In this setting, we expect to get a more accurate picture of the WSD performance of NMT models.

In Section 5, we present a fine-grained investigation of attention distributions of different attention mechanisms. We focus on the process of translating the given ambiguous nouns. Previous studies (Ghader and Monz, 2017; Koehn and Knowles, 2017) have shown that attention mechanisms learn to pay attention to some unaligned but useful context tokens for predictions. Thus, we hypothesize that attention mechanisms distribute more attention to context tokens when translating

[1] Denotes the encoder-decoder attention mechanism in this paper, unless otherwise specified.

[2] More than 2,000 instances in the training set.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 26–35
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64004

ambiguous nouns, compared to when translating other words. To test this hypothesis, we compare the attention weight over ambiguous nouns with the attention weight over all words and all nouns.

In Section 6, we first compare the two different attention mechanisms. Then, we explore the relation between accuracy and attention distributions when translating ambiguous nouns. In the end, we investigate the error distributions over frequency.

Our main findings are summarized as follows:

- We find that WSD is challenging in NMT, and data sparsity is one of the main issues.
- We show that attention mechanisms prefer to pay more attention to the ambiguous nouns rather than context tokens when translating ambiguous nouns.
- We conclude that encoder-decoder attention is not the main mechanism used by NMT models to incorporate contextual information for WSD. Experimental results suggest that models learn to encode contextual information necessary for WSD in the encoder hidden states.
- We reveal that the attention mechanism in Transformers first gradually learns to extract features from the "aligned" source tokens. Then, it learns to capture features from the related but unaligned source context tokens.

## 2 Related Work

Both Rios et al. (2017) and Liu et al. (2018) propose some techniques to improve the translation of ambiguous words. Rios et al. (2017) use sense embeddings and lexical chains as additional input features. Liu et al. (2018) introduce an additional context vector. There is an apparent difference in evaluation between these two studies. Rios et al. (2017) design a constrained WSD task. They create well-designed test sets to evaluate the performance of NMT models in distinguishing different senses of ambiguous words, rather than evaluating the translations of ambiguous words directly. By contrast, Liu et al. (2018) evaluate the translations of ambiguous words but on a common test set. Scoring the contrastive translations is not evaluating the real output of NMT models. In this paper, we directly evaluate the translations generated by NMT models, using *ContraWSD* as the test set.

In NMT, the encoder may encode contextual information into the hidden states. Marvin and Koehn (2018) explore the ability of hidden states at different encoder layers in WSD, while we focus on exploring the attention mechanisms that connect the encoder and the decoder.

Koehn and Knowles (2017) and Ghader and Monz (2017) investigate the relation between attention mechanisms and the traditional word alignment. They find that attention mechanisms not only pay attention to the aligned source tokens but also distribute attention to some unaligned source tokens. In this paper, we perform a more fine-grained investigation of attention mechanisms, focusing on the task of translating ambiguous nouns. We also explore the advanced attention mechanisms in Transformer models (Vaswani et al., 2017).

The encoder-decoder attention mechanisms differ in NMT models. Tang et al. (2018b) evaluate different NMT models, but focusing on NMT architectures. Tang et al. (2018a); Domhan (2018) compare different attention mechanisms. However, there is no detailed analysis on attention mechanisms.

In this paper, we mainly investigate the encoder-decoder attention mechanisms. More specifically, we explore how attention mechanisms work when translating ambiguous nouns.

## 3 Background

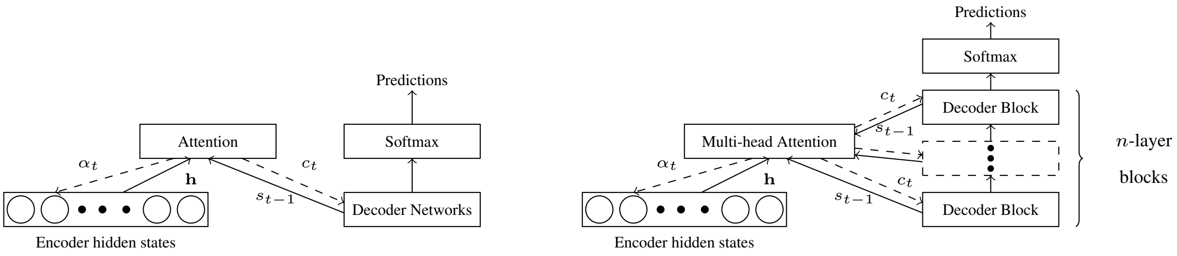
### 3.1 Attention Mechanisms

Attention mechanisms were initially proposed to learn the alignment between source and target tokens by Bahdanau et al. (2015) and Luong et al. (2015), in order to improve the performance of NMT. However, attention mechanisms are different from the traditional word alignment in SMT which learns the hard alignment between source and target tokens. Attention mechanisms learn to extract features from all the source tokens when generating a target token. They assign weights to all the hidden states of source tokens. The more related hidden states are assigned larger weights. Then attention mechanisms feed a *context vector* $c_t$, which is extracted from the encoder, into the decoder for target-side predictions.

We use $\mathbf{h}$ to represent the hidden state set $\{h_1, h_2, \cdots, h_n\}$ in the encoder, where $n$ is the number of source-side tokens. Then $c_t$ is computed by Equation 1:

$$c_t = \alpha_t \mathbf{h} \tag{1}$$

where $\alpha_t$ is the attention vector at time step $t$. $\alpha_t$ is



(a) Vanilla attention mechanism

(b) Advanced attention mechanism

Figure 1: Different attention mechanisms between encoders and decoders in NMT.

a normalized distribution of a score computed by the hidden state set $\mathbf{h}$ and the decoder state $s_{t-1}$, as described by Equation 2:

$$a_t = softmax(score(s_{t-1}, \mathbf{h})) \tag{2}$$

There are different $score()$ functions to compute the *attention vector* $a_t$, including multi-layer perceptron (MLP), dot product, multi-head attention, etc. In this paper, the vanilla attention mechanism employs MLP. The advanced attention mechanism applies multi-head attention with scaled dot product, which is the same as the attention mechanism in Transformer (Vaswani et al., 2017).

Figure 1 illustrates different attention mechanisms. In vanilla attention mechanisms (Bahdanau et al., 2015; Luong et al., 2015), the *context vector* $c_t$ is only fed into the first layer of the decoder networks. Then the single- or multi-layer decoder networks compute from bottom to top to predict target tokens. The vanilla attention mechanisms can only extract the source-side features once, which may be insufficient. Therefore, Gehring et al. (2017) and Vaswani et al. (2017) feed a context vector into each decoder layer. The higher layer could take the result of the previous layer into account when computing the new attention. More recently, Domhan (2018) has shown that multi-layer attention is crucial in NMT models. Moreover, Vaswani et al. (2017) also propose the multi-head attention mechanism. In contrast to the single-head attention, there are multiple attention functions which compute the attention from the linearly projected vectors in parallel. Then, the context vectors from all the heads are concatenated and fed into the decoder networks.

### 3.2 *ContraWSD*

*ContraWSD*[3] from Rios et al. (2017) consists of contrastive translation sets where the human reference translations are paired with one or more contrastive variants. Given an ambiguous word in the source sentence, the correct translation is replaced by an incorrect translation corresponding to another meaning of the ambiguous word. For example, in a case where the English word ‘line’ is the correct translation of the German source word ‘Schlange’, *ContraWSD* replaces ‘line’ with other translations of ‘Schlange’, such as ‘snake’ or ‘serpent’, to generate contrastive translations. To evaluate the performance on disambiguation, contrastive translations are designed not to be easily identified as incorrect based on grammatical and phonological features.

*ContraWSD* is extracted from a large amount of balanced parallel text. It contains 84 different German word senses. It has 7,200 German→English lexical ambiguities and each lexical ambiguity instance has 3.5 contrastive translations on average. All the ambiguous words are nouns so that the WSD is not simply based on syntactic context.

## 4 Evaluation

Instead of using NMT models to score the contrastive translations, we use NMT models to translate source sentences and evaluate the translations of the ambiguous nouns directly. We evaluate two popular NMT models with different attention mechanisms. One is *RNNS2S* with the vanilla attention mechanism, and the other is *Transformer* with the advanced attention mechanism.

We apply *fast-align* (Dyer et al., 2013) to get the aligned translations of ambiguous nouns. To achieve better alignment, we run *fast-align* on both training data and test data which includes reference translations and generated translations. However, for some ambiguous nouns, there is no alignment. We call these ambiguous nouns *filtered*.

There are multiple reference translations for

[3] https://github.com/a-rios/ContraWSD

each ambiguous noun in *ContraWSD*. We additionally add their synonyms[4] into the reference translations as well. The non-reference translations are crawled from the Internet[5].

In addition to the *filtered* nouns, the translations of the ambiguous nouns are classified into six groups, depending on which class (references, incorrect senses, no translation) the translations at aligned/unaligned positions belong to, as described in Table 1. For instance, in *C3*, there is neither a correct nor an incorrect sense at the aligned position. However, there is a reference translation at an unaligned position.

| Group | Aligned | | | Unaligned | | |
|---|---|---|---|---|---|---|
| | Ref. | Incor. | No | Ref. | Incor. | No |
| *C1* | √ | | | | | |
| *C2* | | √ | | √ | | |
| *W1* | | √ | | | √ | √ |
| *C3* | | | √ | √ | | |
| *W2* | | | √ | | √ | |
| *Drop* | | | √ | | | √ |

Table 1: Different groups of translations. *Ref.* denotes the reference translations. *Incor.* represents the incorrect senses. *No* means that there is neither a correct nor an incorrect sense of the ambiguous noun. √ indicates that the translations belong to the reference translations or incorrect senses or neither.

Since the alignment learnt by *fast-align* is not perfect, we also consider the translations at unaligned positions. All the translations in *C1, C2, C3* groups are viewed as correct translations. Thus, the accuracy of an NMT model on this test set is the amount of translations in Group *C1, C2, C3*, divided by the sum of ambiguous noun instances. Formally, $Accuracy = (C1 + C2 + C3)/(C1 + C2 + W1 + C3 + W2 + Drop + Filtered)$, where $C1$, $C2$, $W1$, $C3$, $W2$, $Drop$, and $Filtered$ are the amount of translations in each group.

### 4.1 Experimental Settings

We use the *Sockeye* (Hieber et al., 2017) toolkit, which is based on MXNet (Chen et al., 2015), to train models. In addition, we have extended *Sockeye* to output the distributions of encoder-decoder attention in Transformer models, from different attention heads and different attention layers.

All the models are trained with 2 GPUs. During training, each mini-batch contains 4096 tokens. A model checkpoint is saved every 4,000 updates. We use *Adam* (Kingma and Ba, 2015) as the optimizer. The initial learning rate is set to 0.0002. If the performance on the validation set has not improved for 8 checkpoints, the learning rate is multiplied by 0.7. We set the early stopping patience to 32 checkpoints. All the neural networks have 8 layers. For *RNNS2S*, the encoder has 1 bi-directional LSTM and 6 stacked uni-directional LSTMs, and the decoder is a stack of 8 uni-directional LSTMs. The size of embeddings and hidden states is 512. We apply layer-normalization and label smoothing (0.1) in all models. We tie the source and target embeddings. The dropout rate of embeddings and Transformer blocks is set to 0.1. The dropout rate of RNNs is 0.2. The attention mechanism in *Transformer* has 8 heads.

We use the training data from the WMT17 shared task.[6] We choose *newstest2013* as the validation set, and use *newstest2014* and *newstest2017* as the test sets. All the BLEU scores are measured by *SacreBLEU*. There are about 5.9 million sentence pairs in the training set after preprocessing with Moses scripts. We learn a joint BPE model with 32,000 subword units (Sennrich et al., 2016). There are 6,330 sentences left after filtering the sentences with segmented ambiguous nouns. We employ the models that have the best perplexity on the validation set for the evaluation.

### 4.2 Results

Table 2 gives the performance of NMT models on *newstest*s and *ContraWSD*. The detailed translation distributions over different groups are also provided. *Transformer* is much better than *RNNS2S* in both *newstest*s and *ContraWSD*. Compared to the accuracy of scoring contrastive translation pairs (*Score*), the accuracy of evaluating the translations (*Acc.*) is apparently lower.

There are 8–10% of ambiguous nouns belonging to *Drop* and *Filtered* for both models. We manually checked the translations of sentences with these ambiguous nouns and found that 250 and 206 ambiguous nouns (41%) are translated correctly by *RNNS2S* and *Transformer*, respectively. Our automatic classification failed for two reasons. On the one hand, because the models are trained at subword-level, there are a lot of subwords in the translations. The correctly gener-

[4] Synonyms from WordNet (Miller, 1995)

[5] https://www.linguee.com/german-english

[6] http://www.statmt.org/wmt17/translation-task.html

| Model | *2014* | *2017* | *C1* | *C2* | *W1* | *C3* | *W2* | *Drop* | *Filtered* | *Acc.* | *Score* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *RNNS2S* | 23.3 | 25.1 | 4,560 | 187 | 863 | 81 | 31 | 333 | 275 | 76.27 | 84.01 |
| *Transformer* | 26.7 | 27.5 | 4,982 | 140 | 599 | 85 | 23 | 308 | 193 | 82.26 | 90.34 |

Table 2: Evaluation results of NMT models and the distributions of translations. *2014* and *2017* denote the BLEU scores on *newstest2014* and *newstest2017*, *Acc.* (in %) is short for accuracy. *Score* (in %) is the accuracy using NMT models to score contrastive translation pairs. *Filtered* is the amount of translations that there is no learnt alignment for the ambiguous nouns.

ated translations are subword sequences, and not all the subwords (sometimes even no subword) are aligned to the ambiguous nouns by *fast-align*. On the other hand, the reference translations are all nouns. If the translations are verbs or variants, they are not recognized. If we move these translations into *C1*, the accuracy of the two NMT models will be improved from 76.27% to 80.22%, and from 82.26% to 85.51%, respectively. Thus, attentional NMT models are good at sense disambiguation in German→English, but there is much room for improvement as well.

## 5 Ambiguous Nouns in Attentional NMT

Ghader and Monz (2017) show that there are different attention patterns for words of different part-of-speech (POS) tags, which sheds light on interpreting attention mechanisms. In this section, we investigate the attention distributions over source-side ambiguous nouns.

### 5.1 Hypothesis and Tests

Attention mechanisms not only pay attention to the hidden states at aligned positions but also distribute attention to the hidden states at unaligned positions. The hidden states at unaligned positions can influence the generation of the current token. In general, NLP models disambiguate ambiguous words by means of context words. Thus, for ambiguous nouns, we hypothesize that attention mechanisms distribute more attention to context tokens for disambiguation.

We test our hypothesis via two different comparisons. We use $w_{ambi}$ to denote the average attention weight over the ambiguous nouns and employ $w_{nouns}$ to represent the average attention weight over all nouns[7] (including the ambiguous nouns), while $w_{tokens}$ denotes the average attention weight over all tokens.[8] We first compare $w_{ambi}$ with $w_{tokens}$. As nouns have a more concentrated attention distribution than other word types (Ghader and Monz, 2017), we then compare $w_{ambi}$ with $w_{nouns}$. If $w_{ambi}$ is the smallest, it supports our hypothesis.

The NMT models we evaluated are trained at subword-level. When we compute the attention distributions, we only consider the ambiguous nouns that are not segmented into subwords. To some extent, we therefore conduct an analysis of frequent tokens. We employ the alignment learnt by *fast-align* to find the step of translating the current source token.

Given the attention distribution matrix $M \in \mathbb{R}^{l_s * l_t}$ of a sentence translation, $l_t$ represents the length of the target sentence, while $l_s$ denotes the length of the source sentence. Each column is the attention distribution over all the source tokens when generating the current target token. Each row is the attention distribution over the current source token at all the translation steps. $w$ represents the attention weight over any tokens. If the $i$th source token is aligned to the $j$th target token, then $w = [M]_{ij}$. If a token is aligned to more than one token, we choose the largest attention weight as $w$.[9]

As for Transformer attention mechanisms, there are multiple layers, and each layer has multiple heads. We maximize the attention weights in different heads to represent the attention distribution matrix for each attention layer.[10] We first compute $w_{ambi}$, $w_{nouns}$, and $w_{tokens}$ for each attention layer. Then we average these weights.

### 5.2 Results

As Table 3 shows, $w_{ambi}$ is substantially larger than $w_{tokens}$ in both two models. Even though $w_{nouns}$ is much larger compared to $w_{tokens}$, $w_{ambi}$

[7] We use the TreeTagger (Schmid, 1999) to tag German.

[8] Subword tokens are excluded, which account for 32%.

[9] A source token may be aligned to a set/subset of subword sequences, but the attention mechanism only assigns the corresponding weight to one of the subwords. We select the maximal weight rather than the average weight.

[10] We visualize both the maximal and average attention weights. We find that maximal attention weights are more representative in feature extraction.

is still greater than $w_{nouns}$, especially in *Transformer*. This result is against our hypothesis. That is to say, attention mechanisms do not distribute more attention to context tokens when translating an ambiguous noun. Instead, attention mechanisms pay more attention to the ambiguous noun itself. We assume that the contextual information has already been encoded into the hidden states by the encoder, and attention mechanisms do not learn which source words are useful for WSD.

| Model | $w_{ambi}$ | $w_{tokens}$ | $w_{nouns}$ |
|---|---|---|---|
| *RNNS2S* | 0.63 | 0.48 | 0.62 |
| *Transformer* | 0.74 | 0.57 | 0.69 |

Table 3: Average attention weights over ambiguous nouns, non-subword tokens, and nouns.

Figure 2 demonstrates the average attention weights of the ambiguous nouns, nouns, and non-subword tokens in different Transformer attention layers. In each attention layer, $w_{ambi}$ is always the largest attention weight. It is very interesting that the attention weights keep increasing at lower layers and achieve the largest weight at Layer 5. Then $w_{tokens}$ decreases steadily, while $w_{ambi}$ and $w_{nouns}$ have a distinct drop in the final attention layer. We also re-train a model with 6 attention layers, and we get a figure with the same pattern, but the largest attention weights appear at Layer 4. We will give a further analysis of Transformer attention mechanisms in Section 6.1.

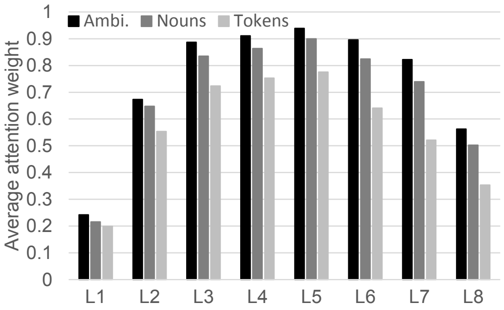


Figure 2: Average attention weights of ambiguous nouns, nouns, and non-subword tokens in different Transformer attention layers.

## 6 Analysis

We first give our analysis of the two different attention mechanisms based on the attention distributions and visualizations. Then, we explore the relation between translation accuracy and attention weight over the ambiguous nouns. In the end, we provide the error distributions over frequency.

### 6.1 Vanilla Attention vs. Advanced Attention

As Table 2 shows, the Transformer model with advanced attention mechanisms is distinctly better than the RNN model with vanilla attention mechanisms. Even though there are differences in the encoder and decoder networks, we focus on the comparison between these two attention mechanisms. Moreover, there is no existing empirical interpretation of the advanced attention mechanisms.

Figure 3 demonstrates the attention distributions of different models when translating ambiguous nouns. For the vanilla attention mechanism in the RNN model, most of the attention weights are relatively uniformly distributed in $[0.5, 0.9)$. While the patterns in advanced attention mechanisms are completely different. In the first layer, most of the attention weights are smaller than $0.1$. The larger attention weights, the fewer instances, except when the weight is larger than $0.9$. In the following layers, the attention weights are getting more and more concentrated in $[0.9, 1)$ until the fifth layer. After the fifth layer, the amount in $[0.9, 1)$ decreases dramatically. We hypothesize that the first few layers are learning the "alignment" gradually. When attention mechanisms finish the "alignment" learning, they start to capture contextual features from the related but unaligned context tokens. In the last layer, the attention is almost equally distributed over all the attention ranges except $(0, 0.1)$. That is to say, for some ambiguous nouns, the weights are large. For the other ambiguous nouns, the weights are small. It indicates that there is no clear attention distribution pattern over ambiguous nouns in the last layer.

Figure 4 shows the average attention weights over word tokens and subword tokens ($w_{subwords}$). In the first five layers, $w_{subwords}$ is clearly lower than $w_{tokens}$ which can be taken to show that attention mechanisms focus on the "alignment" of single word tokens, while $w_{subwords}$ surpasses $w_{tokens}$ from the sixth layer. We conclude that attention mechanisms focus on subwords instead of word tokens. Many words are segmented into multiple consecutive subwords and not all the subwords are aligned to the expected target tokens. Thus, the pattern over subword tokens demonstrates that attention mechanisms are learning to capture context-level features.

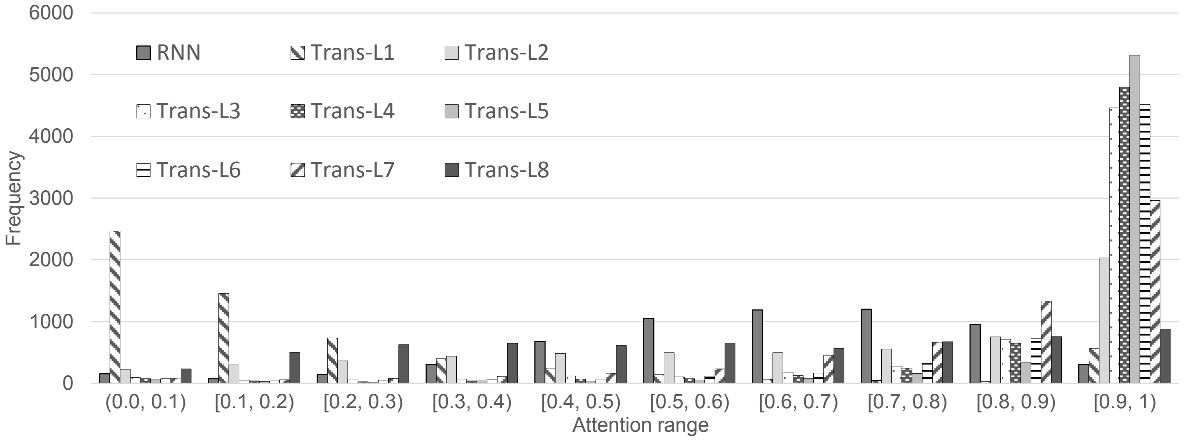


Figure 3: Attention distributions for translating ambiguous nouns from different models. *Trans-L3* denotes the third attention layer in the Transformer model.

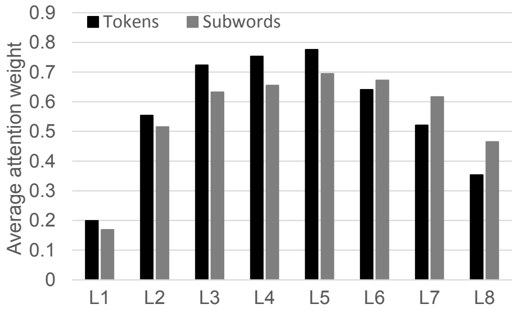


Figure 4: Average attention weights of non-subword tokens and subwords in different Transformer attention layers.

We further validate the hypothesis by visualizing the attention distributions. Table 4 demonstrates the visualization of attention distributions of different attention mechanisms.

‘Stelle’ is an ambiguous noun, whose reference translations are ‘job/position/work’. ‘Stelle’ also has other translations such as ‘location/spot/site’. The context tokens ‘garantiert’ (guarantee) and ‘Leuten’ (people) contribute to disambiguating ‘Stelle’. However, the RNN model could translate ‘Stelle’ correctly but only pays a little attention to ‘Leuten’.

In the first layer, the attention mechanism does not pay attention to the correct source tokens if we only consider the larger attention weights. Then the “alignment” is learnt gradually in the following layers. The attention mechanism could pay attention to all the correct source tokens in the fifth layer. In addition, the attention mechanism could learn to pay attention to the related but unaligned source tokens in the eighth layer. For instance, the attention mechanism also attends to ‘Stelle’ when generating ‘guarantees’, and attends to ‘garantiert’ and ‘Leuten’ when generating ‘job’. These source tokens are not clearly attended to in the fifth layer.

Since the vanilla attention mechanism is only one layer with one head, it does not perform as well as the advanced attention mechanism in learning to pay attention to context tokens. For instance, the attention mechanism in RNN only distributes a little attention to ‘Leuten’ when generating ‘job’.

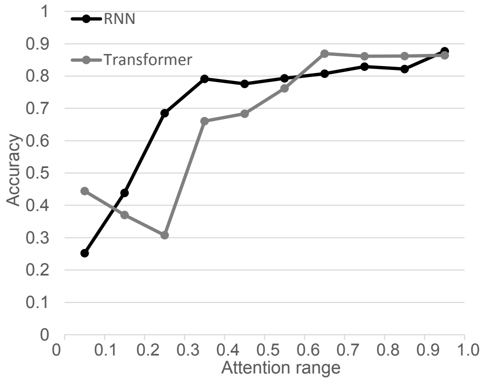
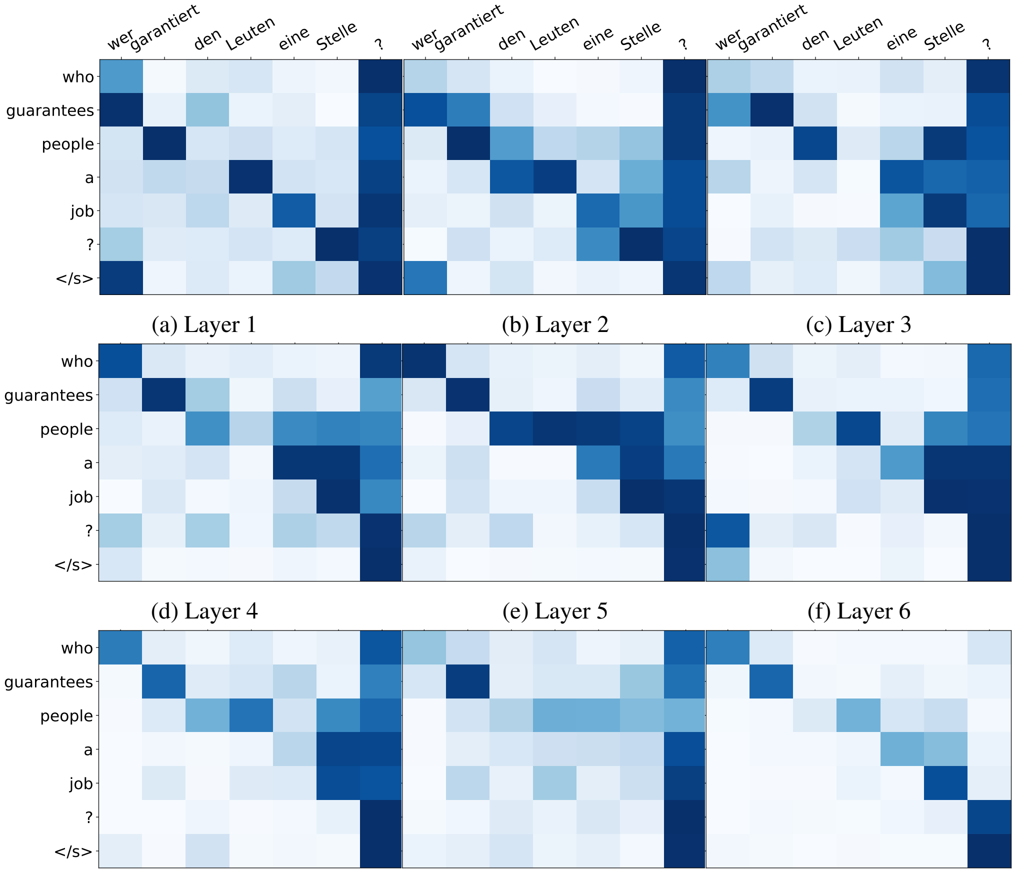


Figure 5: WSD accuracy over attention ranges.

### 6.2 Accuracy and Attention Weights

We explore the relation between WSD accuracy and the attention weights over ambiguous nouns. As the alignment learnt by *fast-align* does not guarantee that each ambiguous noun is aligned to the corresponding translation, we only consider the translations belonging to Group *C1*, *W1*, and *Drop*. Figure 5 shows the WSD accuracy over different attention ranges. Obviously, the accuracy is higher when the attention weight is greater. This



(a) Layer 1 (b) Layer 2 (c) Layer 3

(d) Layer 4 (e) Layer 5 (f) Layer 6

(g) Layer 7 (h) Layer 8 (i) RNN

Table 4: An example of attention visualization (German→English). Each row is the attention distribution over all the source tokens at each time step. Each column represents the attention weight over a source token at all the time steps. Layer 1 to Layer 8 are attention layers in the Transformer model. Each attention layer has 8 heads, and the attention weights in each row are the maximal of all the heads. Thus, the summation of attention weights in each row is larger than 1. Darker blue means larger attention weights.

result further confirms our assumption in Section 5 that the contextual information for disambiguation has been learnt by the encoder. In the attention range $(0, 0.3)$, the small attention weight causes many ambiguous nouns to be untranslated, which results in low WSD accuracy.

### 6.3 Error Distribution

Figure 6 shows the error distributions over absolute frequency (sense frequency in the training set) and relative frequency (sense frequency to source word frequency). The frequency information is given in the test set. It is very clear that most of the errors are in the left bottom corner which are low in both absolute frequency and relative frequency. There are 84.1% and 80.8% errors with an absolute frequency of less than 2000 in *RNN* and *Transformer*, respectively.

Even though the attention mechanism pays a lot

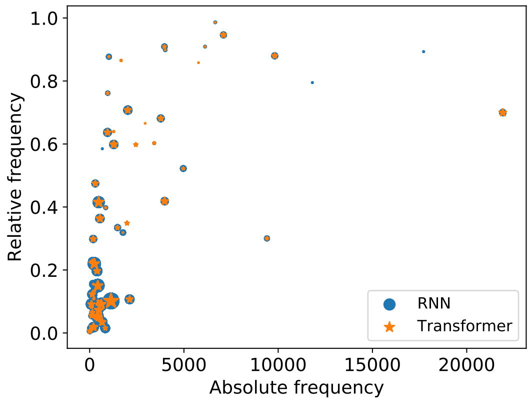


Figure 6: Error distributions over frequency. Absolute frequency is the sense frequency in training set. Relative frequency is the sense frequency in relation to source word frequency. The size of the marker indicates how often the error occurs.

of attention to a low-frequency sense, the model is still likely to generate an incorrect translation. Our evaluation method is different from Rios et al. (2017), but the finding is the same, namely that data sparsity leads to incorrect translations.

## 7 Conclusion

In this paper, we analyze two different attention mechanisms with respect to WSD in NMT. We evaluate the translations of ambiguous nouns directly rather than scoring the contrastive translations pairs, using *ContraWSD* as the test set. We show that the WSD accuracy of these two models is around 80.2% and 85.5%, respectively. Data sparsity is the main problem causing incorrect translations. We hypothesize that attention mechanisms distribute more attention to context tokens to guide the translation of ambiguous nouns. However, we find that attention mechanisms are likely to pay more attention to the ambiguous noun itself. Compared to vanilla attention mechanisms, we reveal that the first few layers in Transformer attention mechanisms learn to “align” source and target tokens, while the last few layers learn to distribute attention to the related but unaligned context tokens. We conclude that encoder-decoder attention is not the main mechanism used by NMT models to incorporate contextual information for WSD. In addition, Section 6.2 has told us that the larger attention weights, the higher WSD accuracy. Tang et al. (2018b) have shown that Transformer models are better than RNN models in WSD because of their stronger encoding ability. These results suggest that NMT models learn to encode contextual information necessary for WSD in the encoder hidden states.

The question how NMT models learn to represent word senses and similar phenomena has implications for transfer learning, the diagnosis of translation errors, and for the design of architectures for MT, including architectures that scale up the context window to the level of documents. We hope that future work will continue to deepen our understanding of the internal workings of NMT models.

## Acknowledgments

We thank all the anonymous reviews who give a lot of valuable and insightful comments. Gongbo Tang is funded by the Chinese Scholarship Council (grant number 201607110016).

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. In *Proceedings of the Workshop on Machine Learning Systems in Neural Information Processing Systems 2015*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Tobias Domhan. 2018. How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1799–1808. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, USA. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1243–1252, Sydney, Australia. The Proceedings of Machine Learning Research.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 30–39, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California, USA.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. 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, Edmonton, Canada. Association for Computational Linguistics.

Frederick Liu, Han Lu, and Graham Neubig. 2018. Handling homographs in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1336–1345. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Rebecca Marvin and Phillip Koehn. 2018. Exploring word sense disambiguation abilities of neural machine translation systems. In *Proceedings of AMTA 2018 (Volume 1: MT Research Track)*, pages 125–131, Boston, USA. Association for Machine Translation in the Americas.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Annette Rios, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19, Copenhagen, Denmark. Association for Computational Linguistics.

Helmut Schmid. 1999. Improvements in part-of-speech tagging with an application to german. In *Natural language processing using very large corpora*, pages 13–25. Springer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Neural Information Processing Systems 2014*, pages 3104–3112, Montréal, Canada.

Gongbo Tang, Fabienne Cap, Eva Pettersson, and Joakim Nivre. 2018a. An evaluation of neural machine translation models on historical spelling normalization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1320–1331. Association for Computational Linguistics.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018b. Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

David Vickrey, Luke Biewald, Marc Teyssier, and Daphne Koller. 2005. Word-sense disambiguation for machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 771–778, Vancouver, Canada. Association for Computational Linguistics.

# Discourse-Related Language Contrasts in English-Croatian Human and Machine Translation

**Margita Šoštarić**[†][*]
[†]University of Zagreb
Dept. of English
margita.sostaric1@ffzg.hr

**Christian Hardmeier**[*] **Sara Stymne**[*]
[*]Uppsala University
Dept. of Linguistics and Philology
first.last@lingfil.uu.se

## Abstract

We present an analysis of a number of coreference phenomena in English-Croatian human and machine translations. The aim is to shed light on the differences in the way these structurally different languages make use of discourse information and provide insights for discourse-aware machine translation system development. The phenomena are automatically identified in parallel data using annotation produced by parsers and word alignment tools, enabling us to pinpoint patterns of interest in both languages. We make the analysis more fine-grained by including three corpora pertaining to three different registers. In a second step, we create a test set with the challenging linguistic constructions and use it to evaluate the performance of three MT systems. We show that both SMT and NMT systems struggle with handling these discourse phenomena, even though NMT tends to perform somewhat better than SMT. By providing an overview of patterns frequently occurring in actual language use, as well as by pointing out the weaknesses of current MT systems that commonly mistranslate them, we hope to contribute to the effort of resolving the issue of discourse phenomena in MT applications.

## 1 Introduction

Every natural language has means of marking elements belonging to the same coreference chain in order to achieve cohesion and coherence in running text. These discourse phenomena are crucial for understanding texts and their misrepresentation harms text intelligibility. Despite their significance, machine translation (MT) systems have been known to struggle with adequately transferring coreference relations from the source to the target language, which is partly due to the great differences in the way languages express these relations. In our approach, we extend the framework introduced by Lapshinova-Koltunski and Hardmeier (2017), who identify discourse discrepancies in parallel data for English and German by predefining and automatically extracting discourse patterns of interest, and then utilize word alignment information to determine which of the extracted phenomena lack the equivalent counterpart in the other language. We use the same procedure to automatically extract phenomena, but extend the methodology to include cases where the phenomenon does have an equivalent construction in the other language, despite the alignment data suggesting that it is more frequently left unaligned.

In this research, we perform an in-depth study of the way in which diverse discourse phenomena are handled in translation from English to Croatian. We investigate both human translation and the output of different types of MT systems. In the first step, we use the extended methodology of Lapshinova-Koltunski and Hardmeier (2017) to extract interesting diverging discourse patterns that commonly occur in the parallel data. While reflections on the relevant linguistic intuitions are given as a reference, the selection of the phenomena chosen for further examination is primarily based on the data obtained from corpora. This makes our approach strongly usage-based and provides ample space for making observations unconstrained by a particular theoretical framework.

In the second step, we construct a dataset with sentences containing challenging discourse phenomena identified in the analysis of human translations. The constructed dataset can be used for further research in the field of corpus linguistics and translation studies, but it is also useful for gaining insight about language contrasts that is of relevance to MT researchers. We therefore use it to test and evaluate the performance of several types of MT systems and to that end devise a weighted error classification, tailored to accommodate the complexity of the problem at hand.

The paper is structured as follows: in Section 2 we explain the motivation for the research and

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 36–48
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64005

in Section 3 we give an overview of related work. In Section 4 we describe the used parallel corpora and present the approach to and findings of their analysis. In Section 5 we describe the MT experiment and our approach to error classification. Section 6 contains a discussion of the obtained results and the paper ends with concluding remarks and ideas for future research in Section 7.

## 2 Motivation

As a South Slavic language, Croatian is a morphologically rich language with relatively free word order. Its pronouns and determiners are grammatically marked for seven cases, three genders and two numbers. Additionally, the forms of determiners and some pronouns show the distinction between animate and inanimate, whereas personal pronouns have long and short variants that reflect emphasis, the choice between them affecting the word order and information flow in the sentence. Croatian is a pro-drop language, meaning that pronouns in the subject position tend to be omitted if the agent can be inferred from other features, such as verbal inflection. In comparison, English is a morphologically less diverse and syntactically stricter language, which suggests that the two languages potentially employ quite different mechanisms to express coreference links.

Apart from the inevitable structural differences, there are several general points of divergence that quickly come to light when handling parallel data for the two languages. First of all, although Croatian has means of expressing the notion of definiteness, it does not have articles, which have a prominent role in the English language. Instead, demonstratives and possessives are sometimes used, as well as definite forms of adjectives and, to a certain extent, restrictive relative clauses. There is also a general tendency to avoid passive constructions and inanimate subjects in Croatian, with these structures commonly rephrased using impersonal verb forms with the reflexive pronoun *se*. Moreover, there is no need for cleft constructions, as information flow can be manipulated through word order, which makes pleonastic pronouns largely redundant in Croatian. Finally, it does not easily create participial constructions, preferring to elaborate the concise English participial expressions into full, finite relative clauses using the relative pronoun *koji*.

## 3 Related Work

The study by Lapshinova-Koltunski and Hardmeier (2017) examines discrepancies in discourse structures for the language pair English-German. The structures are defined as linguistic patterns using part-of-speech and dependency annotation and the discrepancies are identified using alignment information by finding elements with no corresponding structure in the parallel sentence. This approach allows for a corpus-based contrastive analysis, since the discrepancies might be an indication of systematic linguistic differences or examples of explicitation and implicitation phenomena in the translation process. The mentioned study is mostly focused around the former and the authors manually investigate definite articles and pronouns in subject position as the most frequent unaligned patterns in both languages. Through the analysis, they were able to obtain quantitative proof of tendencies regarding, for example, article use in generics and differences in the use of relative clauses.

Although our approach largely follows the above described methodology, Lapshinova-Koltunski and Hardmeier (2017) were hardly the first to recognize the need to address discourse phenomena in translation. Given the immense variety of linguistic phenomena that fall within the scope of the term, research on discourse phenomena in translation has often focused on a limited group of phenomena (e.g. Furkó, 2014; Zinsmeister et al., 2012; Bührig and House, 2004), which frequently have to be studied in reference to particular registers (Kunz and Lapshinova-Koltunski, 2015). Moreover, the pronouncedly language-specific character of their form has led to examinations of explicitation and implicitation of these phenomena in translation (Blum-Kulka, 1986). On a similar note, Meyer and Webber (2013) compare implicitation tendencies in human and machine translation and find that the latter displays more cases where the phenomena are kept in translation. Scarton and Specia (2015) assess the impact of discourse structures on MT quality through quantitative analysis, while Lapshinova-Koltunski (2017) compares human and machine translations to identify and describe variation in the distribution of different cohesive devices.

On the other hand, a variety of approaches have also been proposed to incorporate discursive information in the workflow of MT systems. The approaches of Le Nagard and Koehn (2010), Hardmeier and Federico (2010) and Guillou (2012) are based on the projection of the source side annotation of coreferring pronouns. A number of discourse-oriented pronoun prediction systems, statistical and rule-based, have also been developed for the submission for the DiscoMT shared task (Hardmeier et al., 2015). The systems experimented with different coreference resolution techniques to improve the translation of pronouns. In recent approaches, Voita et al. (2018), Jean et al. (2017), Wang et al. (2017), Tiedemann and Scherrer (2017) and Bawden et al. (2018) all attempt to improve the translation of discourse phenomena using context-aware NMT systems. Although the degree of their success varies, all papers notably report improvement over the baseline systems.

However, the evaluation of these systems remains problematic, as MT evaluation research has typically been focused on providing an overall score for documents, either through automatic metrics like BLEU (Papineni et al., 2002), or through human evaluation, such as the ranking of systems in the WMT evaluations (Bojar et al., 2017). There have been attempts at error analysis where specific errors are identified and classified into typologies (Vilar et al., 2006; Stymne and Ahrenberg, 2012; Comelles et al., 2016), but these classifications usually do not target discourse-related phenomena. Taking a more specific approach to MT evaluation, Burlot and Yvon (2017) describe how test suites can be created and used automatically for the evaluation of MT systems on morphological phenomena, while the test suite PROTEST, developed by Guillou and Hardmeier (2016), enables relative comparisons between MT systems in terms of pronoun translation. Bawden et al. (2018) construct a contrastive test set to evaluate anaphoric pronouns, cohesion and coherence by having NMT systems rank a correct and an incorrect translation of an input sentence, whereas Sennrich (2017) describes a ranking approach for evaluating NMT systems on grammaticality.

Some of the above work has specifically targeted the differences in performance between NMT and SMT (Burlot and Yvon, 2017; Sennrich, 2017). There are also other types of error analysis targeting this difference, e.g. based on post-edits (Bentivogli et al., 2016). For Croatian in particular, Klubička et al. (2017) conducted an error analysis of SMT and NMT systems, finding that the translation of function words in general is considerably improved in NMT. However, they do not present separate results for pronouns or other elements with coreference functions.

## 4 Human Translation Analysis

In this section we give an overview of the used datasets and their preprocessing. We also describe the extraction process and the selected phenomena, along with the observations based on the manual data analysis.

### 4.1 Parallel Corpora

As the use of coreference phenomena varies across different registers and text types, we decided to perform the analysis on corpora from three different domains:

- DGT-TM (Steinberger et al., 2012): EU legal texts, 950K sentences
- SETIMES2 (Tiedemann, 2009): newspaper articles, 200K sentences
- TedTalks (Tiedemann, 2012): prepared speeches, 86K sentences

The three datasets cover an interesting range from very formal, strictly standardized and highly repetitive texts (DGT) to fairly loose and informal translation of speeches (TedTalks). For the purposes of the analyses, English is taken as the source and Croatian as the target language. The corpora were tokenized, tagged for parts of speech and parsed using the pre-trained models for the respective languages developed for the annotation pipeline UDPipe (2017). The parallel data were then aligned at word-level with efmaral (Östling and Tiedemann, 2016), using the grow-diag-final-and heuristic (Koehn et al., 2003).

Relying on the approach of Lapshinova-Koltunski and Hardmeier (2017), we used POS-tags and dependency information to extract a high-recall list of pronouns and determiners in both languages, in order to identify potentially interesting coreference patterns. The main criterion for their extraction was the *pron* or *det* tag, as the original research has found this approach to permit reliable identification of phenomena, even with multi-word units. Similarly to Lapshinova-Koltunski and Hardmeier (2017), we couple the

POS-tags with syntactic information to create linguistic patterns in the format *lemma, POS-tag, dependency label* (e.g. *it, pron, nsubj:pass*) and use word-alignment information to identify the equivalent structure in the other language, if it exists. This gave us a dataset of sentence pairs with indicated coreference phenomena, grouped according to the described linguistic patterns.

Although our approach was largely open and we looked into a variety of phenomena, an initial overview analysis of the data revealed noise both in the output of the tools and in the corpora themselves. As a result, the phenomena chosen for closer examination were selected based on the combination of several factors: the interesting tendencies in their translation identified in the brief overall examination of the data, the tentative interpretation of the frequency of their occurrence across the corpora and the purely practical criterion of having a pattern that enables reliable extraction, meaning that we opted for phenomena which were in most cases correctly handled by the parsing and alignment tools.

### 4.2 Analysis of Discourse Phenomena

This subsection contains the description of the studied phenomena[1] and the observations made in relation to the specific datasets. The number of phenomena occurrences per corpus is shown in Table 1.

**KOJI, det, unaligned.** The high frequency of cases where the relative pronoun *koji* is present in Croatian with no corresponding phenomenon on the English side (*who, whom, whose, which, that*) led us to further examine its use. We separate the phenomenon into two groups, depending on whether the relative pronoun has the function of the subject (nominative case) or object (oblique cases) in the relative clause. A major source of unaligned instances with object function seems to be the omission of *that* in English. In both syntactic functions, *koji* is often introduced as a result of elaboration of participial clauses into finite relative clauses. Interestingly enough, introducing relative clauses seems to be a way of dealing with lexical gaps, as illustrated by the example:

EN: *a resealable bag*

| | | | | | |
|---|---|---|---|---|---|
| *vrećica* | ***koja*** | *se* | *može* | *ponovno* | *zatvoriti* |
| bag | that | REFL | can | again | to seal |

‘a bag that can be sealed again’

Moreover, it is a way of making the concise English phrases more natural and understandable in Croatian:

EN: *women-run entreprises*

| | | | |
|---|---|---|---|
| *poduzeća* | ***koja*** | *vode* | *žene* |
| enterprises | that | run | women |

‘enterprises that are run by women’

Essentially, clauses with *koji* seem to constitute a fairly neutral means of paraphrasing, but their overuse might yield unnecessarily elaborate and clumsy constructions. In SETIMES2 we notice a tendency to resort to such paraphrases in order to maintain a more neutral style:

EN: *the beheaded mother*

| | | | | | |
|---|---|---|---|---|---|
| *majka* | ***koja*** | *je* | *ostala* | *bez* | *glave* |
| mother | who | is | left | without | head |

‘the mother who has lost her head’

Here the entire relative clause could be substituted with the Croatian adjective *obezglavljen*, whose meaning is equivalent to that of ‘beheaded’, but whose use is slightly stylistically marked.

**ARTICLES, det, aligned.** We have already mentioned that Croatian has alternative ways of representing definiteness, the most straightforward example of this being through the use of demonstratives and possessives[2]. We were interested to see whether specific contextual features could be distinguished that make the explicitation of these coreference links necessary. In that respect, the function of articles seems to vary among the corpora: while the DGT deploys a strict coreferencing system to ensure precision, cohesion and consistency, in TedTalks articles are more pronouncedly used for emphasis and achieving immediacy and closeness in delivering a speech in front of an audience. SETIMES2 generally retains definiteness for the purposes of cohesion and boosting the effect of reader engagement by making news appear as more relevant:

EN: *to address* ***the*** *problem, he says...*

| | | | | | |
|---|---|---|---|---|---|
| *kako* | *bi* | *se* | *uspješno* | *nosilo* | *s* |
| in order to | would | REFL | successfully | deal | with |
| ***ovim*** | *problemom,* | *kaže* | *Simitis* | | |
| this | problem | says | Simitis | | |

‘to successfully deal with this problem, says Simitis’

[1] The patterns used to refer to phenomena have the following format: *phenomenon, pos-tag, alignment information*. The last feature specifies whether or not the equivalent structure exists in the other language. At a more specific level, phenomena are defined in reference to the Universal Dependency Treebank labels (Nivre et al., 2015).

[2] The automatic annotation of adjective definiteness was not reliable enough to be used for automatic extraction.

| | KOJIsub | KOJIobj | ARTICLES | ITnsubj | ITexpl | ITpass | ITobj | ITobl/nmod | POSSESSIVES |
|---|---|---|---|---|---|---|---|---|---|
| **DGT** | 19747 | 6606 | 10558 | 8019 | 1576 | 3981 | 3113 | 2395 | 9645 |
| **SETIMES2** | 2844 | 1532 | 8304 | 3801 | 400 | 448 | 1648 | 401 | 6842 |
| **TedTalks** | 618 | 300 | 1758 | 4411 | 185 | 134 | 4919 | 1758 | 3043 |

Table 1: Overall number of occurrences of each phenomenon in the respective language per corpus.

**IT, pron, both.** The semantically vague English pronoun *it* can be used in a variety of functions and roles. Given that our approach is based on the patterns produced by the dependency parser, we generally split the phenomenon according to its syntactic function (subject or object), and then break down the two groups into more fine-grained categories. *It* as the subject is hence analysed according to three different patterns: *it* as the subject of an active clause (nsubj), as the subject of a passive clause (nsubj:pass) and as an expletive (expl). In the first case, the behaviour of *it* generally follows that of other Croatian pronouns, i.e. it is frequently omitted. The two latter cases, by contrast, frequently require paraphrasing of varying extent and level of conventionality in Croatian. These typically entail changing the word order and using impersonal constructions:

EN: ***It** is necessary to make them from scratch.*

| | | | | | |
|---|---|---|---|---|---|
| *Potrebno* | *ih* | *je* | *stvoriti* | *od* | *početka.* |
| Necessary | them | is | to create | from | beginning |

‘It is necessary to make them from scratch.’

In the example, the expletive *it* is dropped and the adjective in singular neuter form is shifted to the initial position in the sentence.

Unfortunately, the diversity of forms of *it* in Croatian makes it a tricky task for word alignment tools, which especially comes to light when *it* is in object position and varies both lexically and grammatically depending on the antecedent. Due to the inability to reliably separate aligned from unaligned instances, we abstracted away from this information in analysing how this phenomenon is handled in translation. For *it* as an object we looked at two phenomena, depending on whether the object is preceded by a preposition (obl/nmod) or not (obj). *It* in object position is more frequently retained in Croatian, which is understandable as it is often required by verb valency.

**POSSESSIVES, det, unaligned.** Finally, we noticed that possessives, especially reflexive possessives, are frequently left out on the Croatian side when their introduction is clumsy or redundant. Notably, possessives are sometimes omitted when there are other clear markers of possession in the sentence, encoded for example by verb inflection or indirect objects:

EN: *it did not deny **my** right to vote*

| | | | | | |
|---|---|---|---|---|---|
| *nije* | *mi* | *uskratila* | *pravo* | *da* | *glasujem* |
| did not | to me | deny | right | to | vote |

‘it did not deny me the right to vote’

The specification of possession in the example above is made redundant by the use of the personal pronoun in dative case *mi*. Similar situations frequently occur in the more informal TedTalks corpus, where personal pronouns in dative case are often introduced to denote a degree of familiarity with the audience. Given the nature of the corpus, there is also a relatively large proportion of cases where the possessives are dropped in phrases that are closely tied to the agent (referring to e.g. one’s body parts or family members). On the other hand, SETIMES2 and DGT are somewhat stricter in style and often omit possessives, an interesting tendency being the omission of reflexive possessives in cataphoric reference:

EN: *Shortly after **their** arrival, the royal couple...*

| | | | | |
|---|---|---|---|---|
| *Nedugo* | *nakon* | *dolaska,* | *kraljevski* | *par* |
| Shortly | after | arrival | royal | couple |

‘Shortly upon arrival, the royal couple’

In the example, the reflexive possessive *svoj* referring to the subject is omitted from the adverbial phrase that precedes it. In the DGT data we also notice the tendency to substitute possessives with explicit noun phrases:

EN: *the value of the procurement over **its** entire duration*

| | | | | |
|---|---|---|---|---|
| *vrijednost* | *nabave* | *tijekom* | *cijelog* | *razdoblja* |
| value | procurement | during | entire | period |
| *trajanja* | *nabave* | | | |
| duration | procurement | | | |

‘the value of the procurement during the entire duration of the procurement’

As can be seen, the noun *nabava* is repeated in the translation instead of using the possessive *njezin*.

## 5 MT Experiment

After analysing the parallel data and identifying interesting tendencies regarding the coreference

| | TRAIN | DEV | PREPROCESSING | CONFIGURATION | BLEU |
|---|---|---|---|---|---|
| **SMT** | 1.23M | 4500 | Standard preprocessing: data tokenized and truecased, max. sentence length 80. | Training and tuning using the Moses default settings, order of the n-gram model: 3. | 33.54 |
| **NMT1** | 1.18M | 4500 | Tokenization, max. sentence length 60, min. word frequency 5. | Encoder: 3-layer bidirectional LSTM, hidden size 500. Decoder: 3-layer LSTM, hidden size 500. | 38.14 |
| **NMT2** | 1.18M | 4500 | Tokenization, max. sentence length 60, individual BPE, min. frequency 5. | Encoder: 3-layer bidirectional LSTM, hidden size 500. Decoder: 3-layer LSTM, hidden size 500. | 36.56 |

Table 2: MT systems – training configurations.

phenomena, we wanted to see how they were handled by different types of MT systems. Using our linguistic patterns, we extracted a subset of sentences, targeting the constructions that are handled differently by the two languages. The number of sentences per phenomenon corresponds to the overall frequency of their occurrence, while the proportion of sentences taken from each corpus roughly reflects the differences in corpus sizes. We added a couple of manually selected examples (cases of lexical gaps and unaligned reflexive pronoun *se* in Croatian) to construct a test set comprising a total of 1899 sentence pairs with indicated phenomena of interest[3]. We have made the dataset publicly available[4].

### 5.1 MT Systems

For the experiment we trained a baseline SMT system and several baseline NMT systems. We used open-source toolkits, the phrase-based SMT package Moses (Koehn et al., 2007) and the OpenNMT toolkit (Klein et al., 2017) respectively, and followed the standard training procedures. The NMT systems were based on a sequence-to-sequence architecture with general attention (Luong et al., 2015) and were trained for 13 epochs. We also experimented with sub-word segmentation with byte pair encoding (Sennrich et al., 2016), trained both individually and jointly, for which 10,000 operations were performed. However, only the two models with the highest BLEU scores were retained for the manual analysis. An overview of the chosen MT systems is given in Table 2[5].

The BLEU scores seem to be in line with what could generally be expected from standard MT systems used on relatively repetitive data, except that the performance of NMT systems slightly drops with the application of byte-pair encoding. This calls for further investigation in the future, with some adaptations possibly needed to be made in the training process. However, the BLEU scores are given only as a reference, as it remains questionable whether this evaluation metric can capture the quality of performance on such specific instances as those that are examined in this study. We hence turn to the manual error analysis.

### 5.2 Error Analysis

For the purposes of the manual analysis, the original human translations are taken as a reference and the order of the machine translations is randomized to reduce bias. The MT output is evaluated only with regard to the relevant antecedent and the syntactic structure containing the specific phenomenon, with the rest of the sentence not being taken into account. Based on our initial data analysis, we devised a classification of errors in terms of translation variation acceptability. The categories used in classification are listed in Table 3. The evaluation was performed by one of the authors, who is a native speaker of Croatian.

To reflect the scalar nature of error severity, we assign a penalty to each error category. This also enables us to produce a provisional score for relative comparison and evaluation of the systems. Some clarification might be needed for categories 4 to 6. Agreement error means that the phenomenon does not agree with the grammati-

[3] Due to the nature of the extraction process, the study is largely focused on intra-sentential phenomena. Although the segmented nature of the artificially constructed test set might be considered a constraint, it is difficult to find an alternative way of testing such a variety of phenomena, while retaining as much data as possible for training.

[4] `http://hdl.handle.net/11234/1-2855`

[5] The test and development sets are kept constant, but the training data used for the two NMT systems had to be further filtered due to technical issues.

| error description | category | penalty |
|---|---|---|
| mistranslation | 1 | 1 |
| phen. misrepresented, unacceptable translation | 2 | 1 |
| different formulation, unacceptable translation | 3 | 1 |
| phen. represented, agreement error | 4 | 0.75 |
| phen. represented, lexical error | 5 | 0.5 |
| phen. represented, grammatical error | 6 | 0.5 |
| phen. misrepresented, unacceptable due to style/register | 7 | 0.25 |
| phen. misrepresented, acceptable in the style/register | 8 | 0 |
| different formulation, acceptable translation | 9 | 0 |
| identical translation | 10 | 0 |

Table 3: MT error classification.

cal categories of its antecedent, whereas the lexical and grammatical errors refer to cases such as antecedent mistranslation or errors in the grammatical form of the surrounding elements contained within the same phrase as the phenomenon. As we are primarily interested in the coreference element, we assign a higher penalty for cases where the coreference link gets lost due to agreement mismatch between the antecedent and the observed phenomenon. On a different note, the choice between errors 2, 7 and 8 sometimes seems particularly subjective, but as we conducted the analysis for three different corpora, it was interesting to retain this fine-grained distinction. As an illustration of the complexity of the categorization task, as well as of our approach to it, we provide an example from the created dataset:

**SOURCE:** ... *modifications for the feed currently legally placed on the market...*

**REFERENCE:**

| | | |
|---|---|---|
| ... *izmjene* | *hrane za životinje* | ***koja*** |
| f.pl.nom | f.sg.gen prep f.pl.acc | f.sg.nom |
| modifications | feed | which |

| | | | | |
|---|---|---|---|---|
| *se* | *trenutačno* | *zakonito* | *stavlja* | *na* |
| refl.pron. | adv | adv | 3.sg | prep |
| REFL | currently | legally | place | on |

| |
|---|
| *tržište...* |
| n.sg.acc |
| market |

‘modifications for the feed which is currently legally placed on the market’

**SMT:**

| | | |
|---|---|---|
| ... *izmjene* | *hrane za životinje* | *trenutačno* |
| f.pl.nom | f.sg.gen prep f.pl.acc | adv |
| modifications | feed | currently |

| | | | |
|---|---|---|---|
| *zakonito* | *stavljeni* | *na* | *tržište...* |
| adv | m.pl.nom | prep | n.sg.acc |
| legally | placed | on | market |

‘modifications for the feed currently legally placed on the market’

**NMT1:**

| | | |
|---|---|---|
| ... *izmjene* | *hrane za životinje* | *već* |
| f.pl.nom | f.sg.gen prep f.pl.acc | adv |
| modifications | feed | already |

| | | | |
|---|---|---|---|
| *pravno* | *stavljene* | *na* | *tržište...* |
| adv | f.sg.gen | prep | n.sg.acc |
| juridically | placed | on | market |

‘modifications for the feed already juridically placed on the market’

**NMT2:**

| | | | |
|---|---|---|---|
| ... *izmjene* | *za* | *hranu za životinje* | ***koje*** |
| f.pl.nom | prep | f.sg.acc prep f.pl.acc | f.pl.nom |
| modifications | for | feed | which |

| | | | | | |
|---|---|---|---|---|---|
| *su* | *trenutačno* | *zakonito* | *stavljene* | *na* | *tržište...* |
| 3.pl | adv | adv | f.pl.nom | prep | n.sg.acc |
| are | currently | legally | placed | on | market |

‘modifications relating to the feed which are currently legally placed on market’

The observed phenomenon here is the unaligned relative pronoun *koji* in subject position, which means we evaluate the translation of the noun phrase whose head noun is *feed*, or *hrana*. The reference translation uses the relative pronoun and an impersonal verb form (*se stavlja*) instead of the participial post-modification. SMT keeps the participial form, which would arguably be an acceptable translation in the DGT corpus (error category 8). However, there is an agreement mismatch between the head noun *hrane* (feminine, singular, genitive case) and the participle *stavljeni* (masculine, plural, nominative case). As the phenomenon present in the reference translation is not represented and there are additional errors which make the translation unacceptable, this is an example of error category 2.

The translation produced by NMT1 uses the correct participial form *stavljene*, but makes inadequate lexical choices in the translation of other elements contained in the phrase, translating *currently* and *legally* by *već* and *pravno* instead of *trenutačno* and *zakonito*, respectively. Regardless of the correct participial form, using the relative clause is generally more acceptable in the translation of this particular sentence, so we treat it as a case of misrepresented phenomenon and opt for a more severe punishment by marking it as error category 2, and not 5. Finally, NMT2 uses the relative pronoun *koji*, but the post-modification does not agree with the head noun in number. It is therefore categorized as error 4. As a side note, all three machine translations also lack the durative aspect, which is one of the morphological properties of

| | total | SMT | NMT1 | NMT2 |
|---|---|---|---|---|
| **DGT** | 931 | 41.78 | 43.29 | 38.78 |
| **SETIMES2** | 628 | 17.36 | 37.9 | 38.85 |
| **TedTalks** | 340 | 11.76 | 30 | 27.65 |

Table 4: Percentage of acceptable translations out of the total number of sentences for each corpus.

| | acceptable | unacceptable | score |
|---|---|---|---|
| **SMT** | 538 | 1361 | 1219.5 |
| **NMT1** | 743 | 1156 | 980 |
| **NMT2** | 699 | 1200 | 1036 |

Table 5: Overall number of acceptable and unacceptable translations and the score based on summed-up penalties.

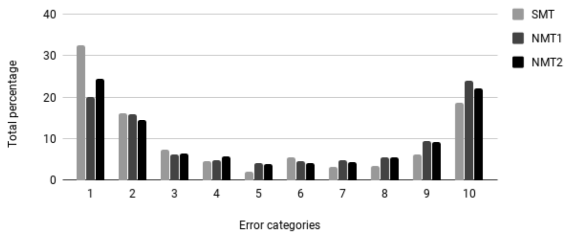


Figure 1: Percentages of error categories for each system.

verbs in Croatian (e.g. *stavljane* instead of *stavljene*), which means that they all belong to error category 6 as well. However, if multiple categories are applicable, we give precedence to the one with the severest penalty so that the overall error scores do not get distorted by single examples.

### 5.3 Results

As already mentioned, the different properties of individual corpora were taken into account in the analysis, but for brevity's sake we focus more on the overall results in the discussion. However, we should point out that all systems generally perform better on the DGT dataset, which is hardly surprising given that it is the largest and most repetitive corpus. As can be seen in Table 4, the variance in performance across corpora is most pronounced in SMT, which produces 42% of acceptable translations for the DGT and only 12% for the TedTalks data.

While for individual phenomena SMT invariably performs best on DGT, there is some variation in the NMT systems, with NMT2 notably performing best on SETIMES2 for all three cases of *it* in subject position and for *koji* as object. Interestingly enough, when it comes to the retention of articles and the omission of possessives, both NMT systems perform best on TedTalks. However, a closer look at the data reveals that the good performance on articles is largely due to NMT producing differently phrased translations (category 9), whereas their performance on possessives is explained by the fact that the informal style and overall proliferation of determiners and pronouns frequently make the retention of possessives seem acceptable (category 8). Finally, we take note of the poor performance of all systems on *it* in obl/nmod function in the TedTalks corpus, with the majority of errors belonging to one of the first three categories and the NMT systems producing the lowest percentage of acceptable translations.

Looking at the overall results, it should be pointed out that the systems generally perform relatively badly on the examined phenomena. As can be seen in Table 5, the systems in total produce more unacceptable than acceptable translations, although the penalty score does seem to loosely reflect the difference in overall translation quality measured by BLEU. For individual phenomena, shown in Table 6, NMT1 consistently performs best, except on possessives and the miscellaneous examples where NMT2 achieves a better score. All systems are most successful in translating *it* as an expletive and passive subject. On the other end of the scale, SMT performs worst on possessives, NMT1 on articles and NMT2 on *it* as object.

In terms of total error counts, SMT produces significantly more complete mistranslations, while NMT2 makes more agreement errors than the other two systems. Both NMT systems also produce more translations that are generally acceptable, but do not fit the given register/style. Overall percentages of individual error categories in the output of each system are shown in Figure 1. We also notice that most cases fall into the extreme ends of the spectrum, i.e. identical translations and mistranslations.

## 6 Discussion

It is often pointed out that NMT systems generally produce more fluent, albeit sometimes inaccurate output compared to SMT. We can therefore hypothesize that the two NMT systems will per-

|  |  | **SMT** |  | **NMT1** |  | **NMT2** |  |
|---|---|---|---|---|---|---|---|
| **phenomenon** | **instances** | **acceptable** | **score** | **acceptable** | **score** | **acceptable** | **score** |
| KOJI, det, subject, unaligned | 237 | 30.8 | 148.5 | 40.51 | 120.5 | 40.93 | 126.25 |
| KOJI, det, object, unaligned | 247 | 33.2 | 143.25 | 46.56 | 102.75 | 43.72 | 110.25 |
| ARTICLES, det, aligned | 327 | 23.85 | 231.25 | 27.83 | 211 | 26.61 | 212.5 |
| IT, pron, nsubj, both | 109 | 33.03 | 64.75 | 44.04 | 50.25 | 39.45 | 57.5 |
| IT, pron, expl, both | 138 | 40.58 | 67.25 | 57.97 | 44.75 | 54.35 | 49 |
| IT, pron, nsubj:pass, both | 137 | 37.23 | 78.5 | 53.28 | 56.25 | 51.82 | 60.5 |
| IT, pron, obj, both | 263 | 22.81 | 180.75 | 32.7 | 148.75 | 25.48 | 165.5 |
| IT, pron, obl/nmod, both | 132 | 27.27 | 86.5 | 36.36 | 74 | 28.03 | 86.25 |
| POSSESSIVES, pron, unaligned | 297 | 21.89 | 209 | 33.33 | 166.75 | 35.69 | 164.25 |
| MISCELLANEOUS | 12 | 8.33 | 9.75 | 58.33 | 5 | 66.67 | 4 |

Table 6: Total scores and percentages of acceptable translations for each system per phenomenon.

form better on unaligned phenomena, especially when the omission or insertion of elements on the target side is more a matter of degree of expression idiomaticity than a strict rule. This is confirmed by our analysis, as NMT systems outperform SMT on all three unaligned phenomena. Moreover, SMT performs worst on possessives, which are generally indeed frequently retained in Croatian, and NMT seems to do a better job at identifying contexts in which they should be left out. As for the relative pronoun *koji* in object position, NMT2 does the best job at recognizing when it is necessary to introduce it on the target side, producing 31.98% of translations identical to the original.

The fluency of NMT could also result in better translations of *it* as an expletive or passive subject, as these instances typically require rephrasing in Croatian. This is confirmed in our analysis to some extent as well, with both NMT systems producing the highest percentage of acceptable translations for these phenomena. However, this is also the case for the SMT system, even if its percentages are much lower, which suggests that the patterns used to paraphrase these two phenomena are fairly standardized in Croatian, and hence frequently occur in the corpora. On the other hand, all systems tend to make mistakes when the rephrasing entails moving a noun into the subject position:

***it*** *is not possible for the controls*

|  |  |  |
|---|---|---|
| *kontrole* | *ne* | *mogu* |
| controls | not | can |

‘the controls cannot’

When it comes to restructuring participial clauses into finite relative clauses using *koji*, the situation is similar. The systems rarely produce the less natural literal translations of participial structures, despite the existence of grammatically equivalent forms in the Croatian language. However, when the translation requires more imaginative paraphrasing, the MT systems in most cases fail to deliver, which highlights their incapability to deal with creative language use and satisfactorily handle lexical gaps. Most cases of such mistranslations, manifested as either omission or retention of the source side element, are noticed for the phenomena of unaligned *koji* and in the small group of miscellaneous examples, which comprises a number of cases chosen specifically to see what the systems will do in situations where the translation and use of coreference phenomena are less straightforward.

For instance, let us consider the innovative phrase *non-carbon-based life*, which in the reference is translated as

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| *život* | *koji* | *se* | *ne* | *bazira* | *na* | *ugljiku* |
| life | which | REFL | not | base | on | carbon |

‘life which is not based on carbon’

and is entirely mistranslated by all three systems. The SMT system leaves the unknown word in source language, misinterprets the dependency relations and substitutes the relative clause with an impersonal verb construction with *se*:

|  |  |  |  |  |
|---|---|---|---|---|
| *non-carbon* | *se* | *temelje* | *na* | *životu* |
| non-carbon | REFL | based | on | life |

‘non-carbon are based on life’

Both NMT systems leave out the entire unknown part and translate the phrase only as *život* (‘life’).

The systems also fail to cope with idiomatic expressions, frequently omitting or producing word-for-word translations for idiomatic uses of *it* in object position (e.g. *make it, get it*). The translation of multi-word units is another well-known stumbling block of MT systems, but this particular discourse phenomenon seems to be problematic for another reason, and that is the already mentioned diversity of grammatical forms this pronoun can take in the object position in Croatian. Incidentally, *it* in object position is the phenomenon for which all three systems produce the largest percentage of agreement errors: well above 20% of errors made by the systems on this phenomenon belong to category 4, compared to the usual average of around 3% of agreement errors produced in the translation of other phenomena. Finally, the relative performance of all three systems lies closest in the case of aligned articles, but that is because all systems perform poorly, probably due to the very strong tendency not to translate these elements that permeate the English side of the corpus.

## 7 Conclusion and Future Work

In this paper, we apply the usage-based approach of Lapshinova-Koltunski and Hardmeier (2017) for automatic identification of unaligned patterns linked to discourse-related language discrepancies, and extend it to also include cases of interesting aligned phenomena. We focus on pronouns and determiners in two structurally different languages, English and Croatian, and study them in parallel corpora pertaining to three different registers. We were able to distinguish tendencies both at the general level (e.g. the omission of reflexive possessives in cataphoric position in Croatian) and at corpus-specific levels (e.g. the stricter regulation of representation of definiteness in the DGT corpus). We find that the data-driven nature of the approach makes it a useful framework for linguistic and translation studies, as it hardly makes any initial assumptions about the behaviour of phenomena.

The observations obtained from the parallel data analysis were used to pinpoint interesting linguistic patterns in the two languages, and we further study the way they are handled in MT. To that end, we trained several statistical and neural MT systems and constructed a test set targeting the challenging linguistic expressions. The test set has been made publicly available for further research. We devised a relatively fine-grained classification of errors to evaluate system performance and assigned a penalty to the different error categories in order to facilitate the comparison and ranking of systems in terms of translation acceptability. We provide insights for these diverse extracted phenomena both with regard to the different registers and to the general performance of several MT systems.

Overall, all systems seem to perform unsatisfactorily, especially so on the TedTalks corpus, which is smallest in size as well as linguistically informal and diverse. On the other hand, insofar as better handling of unaligned phenomena can be interpreted as a reflection of translation fluency, NMT systems seem to outperform SMT by producing a higher percentage of acceptable translations in cases which involve standard patterns of paraphrasing and the introduction/omission of coreference elements on the target side. However, all MT systems fall short when it comes to more creative language use, such as handling lexical gaps or idiomatic expressions. Our analysis highlights the complexity of the issue and offers an approach through which further insights can be obtained with a view to improve the translation of coreference phenomena. Lastly, we would like to point out that the research included Croatian, a language that is both under-resourced and under-researched in the field of MT. We also believe that many of the insights for English–Croatian could carry over to other closely related Slavic languages.

As part of future work it would be interesting to investigate other coreference phenomena, and experiment with basing the extraction patterns on some other linguistic features, such as pronoun function (cf. Guillou et al., 2014). As for MT system applications, our manual analysis suggests that the MT systems for this language pair are generally in need of some improvement to better support the study of such specific phenomena, despite obtaining reasonably high BLEU scores. Further inquiry into why the system performance dropped with the application of byte-pair encoding would certainly be advisable and experimenting with different architectures, notably the Transformer (Vaswani et al., 2017), would be desirable. Future work might also include attempts at integrating the output of coreference annotation systems in the workflow of MT systems, in order to make them more attuned to the translation of discourse phenomena.

## Acknowledgements

Christian Hardmeier received funding from the Swedish Research Council under grant 2017-930. We acknowledge the computational resources provided by CSC in Helsinki through NeIC-NLPL (www.nlpl.eu).

## References

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313. Association for Computational Linguistics.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267.

Shoshana Blum-Kulka. 1986. Shifts of cohesion and coherence in translation. In Juliane House and Shoshana Blum-Kulka, editors, *Interlingual and intercultural communication: Discourse and cognition in translation and second language acquisition studies*, pages 17–35. Tübingen: Günter Narr Verlag.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Franck Burlot and François Yvon. 2017. Evaluating the morphological competence of machine translation systems. In *Proceedings of the Second Conference on Machine Translation*, pages 43–55. Association for Computational Linguistics.

Kristin Bührig and Juliane House. 2004. Connectivity in translation transitions from orality to literacy. *Multilingual communication*, 3:87–114.

Elisabet Comelles, Victoria Arranz, and Irene Castellón. 2016. Guiding automatic MT evaluation by means of linguistic features. *Digital Scholarship in the Humanities*, 32(4):761–778.

Bálint Péter Furkó. 2014. Perspectives on the translation of discourse markers: A case study of the translation of reformulation markers from English into Hungarian. *Acta Universitatis Sapientiae, Philologica*, 6(2):181–196.

Liane Guillou. 2012. Improving pronoun translation for statistical machine translation. In *Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–10. Association for Computational Linguistics.

Liane Guillou and Christian Hardmeier. 2016. Protest: A test suite for evaluating pronouns in machine translation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia. European Language Resources Association (ELRA).

Liane Guillou, Christian Hardmeier, Aaron Smith, Jörg Tiedemann, and Bonnie Webber. 2014. ParCor 1.0: A parallel pronoun-coreference corpus to support statistical MT. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Christian Hardmeier and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In *IWSLT (International Workshop on Spoken Language Translation)*, pages 283–289, Paris, France.

Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, and Mauro Cettolo. 2015. Pronoun-focused MT and cross-lingual pronoun prediction: Findings of the 2015 DiscoMT shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Filip Klubička, Antonio Toral, and Víctor M. Sánchez-Cartagena. 2017. Fine-Grained Human Evaluation of Neural Versus Phrase-Based Machine Translation. *The Prague Bulletin of Mathematical Linguistics*, 108:121–132.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. 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.

Kerstin Kunz and Ekaterina Lapshinova-Koltunski. 2015. Cross-linguistic analysis of discourse variation across registers. *Nordic Journal of English Studies*, 14(1):258–288.

Ekaterina Lapshinova-Koltunski. 2017. Cohesion and translation variation: Corpus-based analysis of translation varieties. In Katrin Menzel andEkaterina Lapshinova-Koltunski and Kerstin Kunz, editors, *New perspectives on cohesion and coherence*, pages 105–130. Berlin: Language Science Press.

Ekaterina Lapshinova-Koltunski and Christian Hardmeier. 2017. Discovery of discourse-related language contrasts through alignment discrepancies in English-German translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 73–81.

Ronan Le Nagard and Philipp Koehn. 2010. Aiding pronoun translation with co-reference resolution. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 252–261. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Thomas Meyer and Bonnie Webber. 2013. Implicitation of discourse connectives in (machine) translation. In *Proceedings of the Workshop on Discourse in Machine Translation*, pages 19–26.

Joakim Nivre, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Yoav Goldberg, Jan Hajič, Jenna Kanerva, Veronika Laippala, Alessandro Lenci, Teresa Lynn, Christopher Manning, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Maria Simi, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. 2015. Universal dependencies 1.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Carolina Scarton and Lucia Specia. 2015. A quantitative analysis of discourse phenomena in machine translation. *Discours. Revue de linguistique, psycholinguistique et informatique. A journal of linguistics, psycholinguistics and computational linguistics*, 16.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? Assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. 2012. DGT-TM: A freely available translation memory in 22 languages. In *Proceedings of the 8th international conference on Language Resources and Evaluation (LREC-2012)*, Istanbul, Turkey.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Sara Stymne and Lars Ahrenberg. 2012. On the practice of error analysis for machine translation evaluation. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC-2012)*, Istanbul, Turkey.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Jörg Tiedemann and Yves Scherrer. 2017. Machine translation with extended context. In *Proceedings of the 3rd Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC-2012)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

David Vilar, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. 2006. Error analysis of machine translation output. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006)*, pages 697–702, Genoa, Italy.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274. Association for Computational Linguistics.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831. Association for Computational Linguistics.

Heike Zinsmeister, Stefanie Dipper, and Melanie Seiss. 2012. Abstract pronominal anaphors and label nouns in German and English: Selected case studies and quantitative investigations. *Translation: Computation, Corpora, Cognition*, 2(1).

# Coreference and Coherence in Neural Machine Translation: A Study Using Oracle Experiments

**Dario Stojanovski** **Alexander Fraser**
Center for Information and Language Processing
LMU Munich
{stojanovski,fraser}@cis.lmu.de

## Abstract

Cross-sentence context can provide valuable information in Machine Translation and is critical for translation of anaphoric pronouns and for providing consistent translations. In this paper, we devise simple oracle experiments targeting coreference and coherence. Oracles are an easy way to evaluate the effect of different discourse-level phenomena in NMT using BLEU and eliminate the necessity to manually define challenge sets for this purpose. We propose two context-aware NMT models and compare them against models working on a concatenation of consecutive sentences. Concatenation models perform better, but are computationally expensive. We show that NMT models taking advantage of context oracle signals can achieve considerable gains in BLEU, of up to 7.02 BLEU for coreference and 1.89 BLEU for coherence on subtitles translation. Access to strong signals allows us to make clear comparisons between context-aware models.

## 1 Introduction

Neural Machine Translation (NMT) (Bahdanau et al., 2015) is a state-of-the-art approach to MT. Standard NMT models translate an input language sentence to an output language sentence, and do not take into account discourse-level phenomena. Cross-sentence context has already proven useful for language modeling (Ji et al., 2015; Wang and Cho, 2016) and dialogue systems (Serban et al., 2016). It has also been of interest in Statistical Machine Translation (SMT) research (Hardmeier, 2012; Hardmeier et al., 2013; Carpuat and Simard, 2012), and NMT research (Wang et al., 2017; Jean et al., 2017; Tiedemann and Scherrer, 2017; Bawden et al., 2018; Tu et al., 2017; Voita et al., 2018).

Two important discourse phenomena for MT are coreference and coherence. Pronominal coreference relates to the issue of translating anaphoric pronouns and is tackled in several works (Guillou, 2016; Hardmeier and Federico, 2010; Le Nagard and Koehn, 2010) and is the central motivation for the DiscoMT shared task on cross-lingual pronoun prediction (Loáiciga et al., 2017). Coherence on the other hand, is important for producing consistent and coherent translations throughout a document, especially for domain-specific terminology (Carpuat, 2009; Ture et al., 2012; Gonzales et al., 2017) and it is helpful to properly disambiguate polysemous words. Modeling discourse-level phenomena for MT is a challenging endeavor because of difficulties in acquiring relevant linguistic signals. Measuring the effect of discourse-level phenomena with automatic metrics such as BLEU is also difficult as pointed out by Hardmeier (2012).

In this paper, we address these issues by proposing several oracle experimental setups for evaluating the effect of coreference resolution (CR) and coherence in MT. Oracle experiments provide strong linguistic signals that enable strongly visible effects on BLEU scores, thus alleviating the difficulty of using BLEU to evaluate discourse-level phenomena in MT. Oracles highlight the capability of NMT systems to use context (which we call context-aware NMT) and to handle different discourse-level phenomena. They provide a variety of scenarios that can easily be set up for any domain, dataset or language pair, unlike discourse-specific challenge sets (Bawden et al., 2018) which must be manually created. Furthermore, strong linguistic signals from oracles enable us to easily study how the models use context.

Our primary task is translating subtitles from English to German. Subtitles provide for a reasonable diversity of topics necessary for testing coherence. They also contain a large amount of short, informal and conversational text, where anaphoric pronouns are very important. We study coreference by aiding pronoun translation and coherence

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 49–60
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64006

by providing disambiguation signals for translation of polysemous words. The oracles are automatically created and targeted for each discourse phenomenon. We additionally include a previous target sentence oracle, where the context consists of the previous target sentence, as a more generic way of including context. This is an interesting oracle, but this scenario is actually also beneficial for online post-editing, because the gold standard previous target sentence is available there.

We propose a simple, yet effective extension to standard RNN models for NMT (which we refer to as NMT(RNN)) which models context by employing attention over word embeddings only. We compare it against a standard NMT(RNN) model working on a concatenation of consecutive sentences (Tiedemann and Scherrer, 2017). Additionally, we evaluate the Transformer (Vaswani et al., 2017) and propose a context-aware NMT(Transformer) extension. Our oracles allow us to compare the context-aware NMT models with the baselines and make strong conclusions. Moreover, we study how comparable oracles are with the challenge sets proposed by Bawden et al. (2018) by analyzing the performance of our context-aware model with both approaches. Finally, we conduct a qualitative study and show the inner workings of context-aware models under different oracle settings.

**Contributions:** (i) We modify the data using an oracle experimental setup in order to accommodate evaluating coreference and coherence in NMT. (ii) Our evaluation is independent of carefully constructed challenge sets, and can easily be transferred across language pairs and domains. (iii) Results clearly show context-aware NMT(RNN) and NMT(Transformer) can improve performance over NMT models without access to context. (iv) We empirically analyze the pros and cons of the major approaches to context-aware NMT and explain how different modeling decisions interact with different discourse phenomena. (v) We present the trade-offs in modeling power versus speed that are important when considering multiple sentences of context.

## 2 Oracle Signals for Coreference and Coherence

Acquiring clean and strong context signals is a difficult challenge and previous work has not proposed a way to do this on a larger scale. In our work, we use oracles, where the context signals are strong and allow us to carry out clear analysis. We define three oracles which differ based on the context supplied to the model.

First, we define the previous target sentence oracle where the context is the gold standard previous target sentence. Second, we define the coreference or pronoun oracle where we simulate perfect knowledge of gender and number for pronoun translation. Finally, we define the coherence or more specifically, the repeated words oracle where we help in identifying polysemous words and providing the correct signal for disambiguation.

Each of these oracles is accompanied by a fair and a noisy oracle experimental setup. For the fair setup, we obtain the linguistic signals in a realistic way without having access to any target side knowledge. In the noisy oracle setups, we add additional target side information to the oracle signals. This additional information is not necessarily relevant to the specific problem at hand (coreference or coherence) and it is used to test the robustness of the models to identify the proper signals.

The oracle datasets are created in an automatic way. We only need to manually define the list of pronouns that will be taken into consideration in the coreference oracle.

**Oracle** Table 1 shows samples from our oracle setup. For each example we show the context, original source sentence, our modified oracle sentence and the target sentence. The first two examples show coreference (pronoun) oracle samples, while the third one a coherence (repeated words) oracle sample. The text in brackets shows which is the counterpart repeated target word or the gender of the noun the pronoun is referencing. It is not explicitly provided to the models. The text preceding the special token *!@#$* in the oracle examples is the input to the context part of the architecture.

For coreference, we aid the model with pronoun translation as can be seen in example (c). In this case, *it* refers to *Roman* (meaning *novel*), which is apparent in the previous sentence (a). Without this information the model will have difficulties generating the proper translation *er* (the German masculine pronoun agreeing with *Roman*).

When creating the pronoun oracle setup, we do not utilize the context sentence. Instead, we just consider the current source and corresponding target sentence. If both sentences contain at least one pronoun in their respective languages, we mark the source pronouns with XPRONOUN and insert the target pronouns in the context of the main sentence, as in example (c).

The example shows that the context provides access to perfect knowledge of the coreferent, which in turn tells us the number and gender. However, the models still need to learn to use the correct pronouns. As we can see in example (g), there may be multiple pronouns in the context. Since (g) is an imperative sentence, *Sie* does not have a pronoun counterpart in the source and it is used in conjunction with the German verb for *use*.

Example (k) shows how we model the coherence phenomenon by using repeated words. Given the English word *source* in a sentence without helpful context, it would be impossible to disambiguate between two possible translations of the word: *Quelle* (a source of a fountain or figuratively the source of information) or *Ursprung* (origin, where something originates from). However, we see that the previous sentence (i) contains the relevant information to select the correct translation of the English *source*. The word *source* is present in the previous and current source sentence and *Ursprung* is present in the previous and current target sentence. When we find at least one repeated word on both the source and target side, we mark the source word with a special token XREP and the repeated target word is used as context to the main source sentence. The intuition here follows previous work (Tu et al., 2017) where past translation decisions are used for disambiguation. This oracle is admittedly weaker than the coreference one since it relies on the assumption that a polysemous word has already been seen in the text. However, if a word occurs in two consecutive sentences, it is likely that it will have the same translation.

For the previous target sentence oracle, we use the gold standard previous target sentence as context and don't modify the main source sentence. We also setup experiments with 2 and 3 previous target sentences as context.

*context sentence*
(a) Let me summarize the novel[masculine] for you.
*source sentence*
(b) It presents a problem
*pronoun oracle sample*
(c) er[masculine] !@#$ XPRONOUN It presents a problem.
*target sentence*
(d) Er präsentiert ein Problem.

---

*context sentence*
(e) But you have a charm[masculine] everyone else here seems to respond to.
*source sentence*
(f) Use it. OK, sport?
*multiple pronoun oracle sample*
(g) Sie ihn[masculine] !@#$ Use XPRONOUN it. OK, sport?
*target sentence*
(h) Setzen Sie ihn ein.

---

*context sentence*
(i) When dealing with a crisis everyone knows you go right to the source[Ursprung].
*source sentence*
(j) God the source is pretty.
*repeated words oracle sample*
(k) Ursprung !@#$ God the XREP source is pretty.
*target sentence*
(l) Mann, so ein hübscher Ursprung.

Table 1: Coreference and coherence oracle samples. For detailed explanation of the examples, refer to Section 2.

**Fair** For the fair coreference setup, we attempt to acquire gender and number knowledge by using a coreference resolution tool, namely CorefAnnotator from Stanford CoreNLP[1] (Clark and Manning, 2016a,b). We run the model on entire documents. We only modified sentences that contain a pronoun which has an antecedent in the previous source sentence. Consequently, the pronoun is marked and the antecedent is inserted into the context of the given sentence. In this way, we don't utilize any target side knowledge.

For the fair coherence experiment, we don't have access to target side information and we just put special emphasis on words that are polysemous candidates. As a result, we only use repeated source words. A repeated word is marked in the main sentence and it is used as context.

For the fair previous sentence experimental setup, we use the same models trained on the previous target sentence oracle setup, but evaluate them by translating the previous source sentence with a baseline model and using this translation as context. Additionally, we train models where the previous sentence is from the source side.

**Noisy oracles** In order to test the robustness of context-aware models, we define noisy coreference oracles. We use the same approach as in the oracle, but the previous gold standard target sentence is added at the beginning of the context (which already contains the target side pronouns).

We also define noisy oracles for coherence. In this case, this is achieved by marking repeated source words and marking repeated target words in the previous target sentence and using the modified previous target sentence as context.

[1] https://stanfordnlp.github.io/CoreNLP

## 3 Related Work

Bawden et al. (2018) is a recent work with similarities to ours. They look at the scores computed by context-aware models using challenge sets, by comparing model scores on two perfect target language sentences differing only on a single choice of, e.g., gender for a pronoun, and providing two different contexts to try to obtain, e.g., masculine in the first case and feminine in the second case.

Like Bawden et al. (2018), we provide a focused evaluation on coherence and coreference, but unlike their work, we do not depend on manually created datasets. Our simple oracles are a strong alternative to manually constructed challenge sets, as we can easily have a more diverse experimental setup (our oracles can be defined for different languages, domains and datasets with little effort).

Several approaches have been proposed for context-aware NMT that utilize a separate mechanism to handle extra-sentential information. Wang et al. (2017) integrate cross-sentence context using gates in the decoder, which control information flow between the cross-sentence context and the current decoder state. However, the context representation is fixed at each decoding time step, while the model needs to focus on different parts of the context. Tu et al. (2017) propose a caching mechanism that stores previous translation decisions. As a result, this approach fails to take into account CR as stored translation decisions can't be used to address this phenomenon. Jean et al. (2017) and Bawden et al. (2018) propose methods using a separate RNN-based context encoder. Tiedemann and Scherrer (2017), propose concatenating the preceding sentence, both on source and target side and then using a standard NMT model. These approaches are computationally expensive. They either have an extra RNN-based encoder (Jean et al., 2017; Bawden et al., 2018) or work on very long sentences (Tiedemann and Scherrer, 2017).

A recent work by Voita et al. (2018) proposed a context-aware Transformer model and provided an analysis of anaphora resolution in MT. Their proposed model is conceptually similar to our NMT(Transformer) model, differing in that the context is integrated in the encoder unlike our model which does it in the decoder.

We propose a simple NMT(RNN) model that only uses attention to encode the context and integrates it with a gating mechanism (Wang et al., 2017). It provides for a better computational efficiency compared to models employing an extra RNN-based encoder. We also propose a context-aware Transformer model. In the experiments, we compare our models against a concatenation NMT(RNN) and NMT(Transformer) model (Tiedemann and Scherrer, 2017).

## 4 Context-Aware Models

### 4.1 Lightweight context-aware NMT(RNN) model

In this paper, we introduce a new lightweight context-aware model based on the attention encoder-decoder model proposed by Bahdanau et al. (2015). We introduce this context-aware model to compare against the proposed model by Tiedemann and Scherrer (2017) as an alternative approach to handling context.

The encoder part of the model, takes the source sentence $X = (x_1, x_2, \ldots, x_{T_x})$ and generates a set of annotation vectors $\{h_1, h_2, \ldots, h_{T_x}\}$ where $h_i = \left[\overrightarrow{h}_i; \overleftarrow{h}_i\right]$. $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ are the $i$-th hidden states from the forward and backward recurrent networks respectively. The decoder generates one target symbol $y_i$ at a time by computing the conditional probability $p(y_i|y_1, y_2, \ldots, y_{i-1}, x) = f(y_{i-1}, s_i, c_i)$ where $c_i$ represents the attention weighted sum of annotation vectors and is computed as in (Bahdanau et al., 2015). Unlike previous approaches that model context by employing an RNN-based encoder (Jean et al., 2017; Bawden et al., 2018), we propose to utilize the capability of the attention mechanism only. This provides for better computational efficiency, thus allowing the model to exploit larger context at a lower computational cost.

The context sentence is given as a sequence of $X^c = (x_1^c, x_2^c, \ldots, x_{T_x^c}^c)$. We map the tokens to the corresponding word embeddings $w_i^c$. We share all embeddings across the model, including the context ones. The attention on the cross-sentence context is conditioned on the previously generated token $y_{i-1}$ current candidate decoder state $s_{i-1}$ and attention weighted main sentence representation $c_i$. Formally, the context sentence representation is computed as $c_i^c = \sum_{j=1}^{T_x^c} \beta_{ij} w_j$ where $\beta \propto exp(f_{att}^c(y_{i-1}, s_{i-1}, w_j, c_i))$.

We integrate the context representation using a gating mechanism (Wang et al., 2017) which controls the flow of information between the current decoder state and the context representation. which is computed as $g = f_g(y_{i-1}, s_{i-1}, c_i, c_i^c)$.

The final decoder representation is computed as $s_i = f_c(y_{i-1}, s_{i-1}, c_i, g \otimes c_i^c)$.

### 4.2 Transformer context-aware model

The Transformer (Vaswani et al., 2017) is an encoder-decoder architecture which fully relies on attention. The encoder layers have two main components, a multi-head self-attention and a position-wise fully-connected feed-forward network. Each of these components is followed by a residual connection. In the self-attention sublayer, each word from the input sentence acts as a query, key and value when computing the attention. Each attention head uses the queries and keys to compute a dot product to which a softmax is applied in order to get the attention weights to score the values. Consequently, the representation of each word depends on all the others. The final representation is generated by concatenating the output of the separate attention heads and inputting it to the feed-forward network. The decoder on the other hand, has three sublayers. It starts by applying masked self-attention which is then used to compute multi-head attention over the encoder representation. This is then used as input to a feed-forward network as in the encoder.

The proposed context-aware model in this paper is built as an extension to the standard Transformer. All embeddings including the context embeddings are shared across the model. We modify the encoder by sharing the parameters for the multi-head self-attention for the main and context sentence. However, we don't share the feed-forward network after the self-attention.

The standard decoder computes a multi-head attention $c_i$ over the main encoder representation using the output from the masked self-attention $c_i^m$. We add an additional multi-head attention over the context representation $c_i^c$ as well. Before computing the context attention, the output of the masked self-attention is projected using a feed-forward network. The main and context multi-head self-attention representations are merged using a gating mechanism as $s_i = g_i \otimes c_i + (1 - g_i) \otimes c_i^c$ where $g_i = \sigma(W_e c_i + W_c c_i^c + W_m c_i^m)$.

## 5 Experiments

We train our models on OpenSubtitles2016 En-De with $\approx$ 13.9M parallel sentences. The development and test set consist of 6 and 7 documents randomly sampled from the dataset, containing 3172 and 4627 sentences respectively. In the coreference oracle setup $\approx$ 7.8M training samples were modified and added the appropriate context, while in the coherence setup only $\approx$ 0.8M. The remaining samples are unchanged and have no context.

We apply tokenization, truecasing and BPE splitting computed jointly on both languages with 59500 operations. All sentences with length above 60 tokens are discarded. Batch size is 80. All embeddings are tied (Press and Wolf, 2017) including the ones in the context part of the architecture. Dropout (Gal and Ghahramani, 2016) of 0.2 is applied and 0.1 on the embeddings. We apply layer (Ba et al., 2016) and weight normalization (Salimans and Kingma, 2016). The models are trained with early-stopping based on the development set's cost. We report BLEU score on detokenized text.

Our RNN-based model is implemented as an extension to Nematus[2] (Sennrich et al., 2017). We used the Sockeye[3] (Hieber et al., 2017) implementation of the Transformer. For the Transformer we use hyper-parameters as similar as possible to the ones in the Nematus models. We additionally use label smoothing of value 0.1. Both, the baseline and context-aware model have 4 layers. We didn't do any special hyper-parameter tuning for the context-aware models, so further performance improvements are possible. The datasets and the source code for our context-aware models are publicly available[4].

## 6 Experimental Results

### 6.1 Previous target sentence oracle

In this section, we discuss the effect of using context in context-aware NMT. In Table 2 we show the results for the three different oracle setups. Experiment (1a) shows that a baseline NMT(RNN) model obtains 28.57 BLEU on the test set. The NMT(Transformer) baseline (1b) on the other hand, achieves 29.53 BLEU. Using the gold standard previous target sentence as context, provides for 1.32 BLEU improvement on the test for our context-aware NMT(RNN) model (2a) and 1.78 BLEU for the concatenation NMT(RNN) model (3a). Our proposed context-

[2] https://github.com/EdinburghNLP/nematus

[3] https://github.com/awslabs/sockeye

[4] http://www.cis.uni-muenchen.de/~dario/projects/oracles

aware NMT(Transformer) model (2b) also improves upon the baseline, but only by 0.6 BLEU, and the concatenation model (3b) closely follows the RNN model, adding 1.49 BLEU.

We also evaluate the usefulness of larger context. Using the previous 2 (6a) and 3 (7a) sentences consistently adds ≈ 0.6 BLEU with the concatenation NMT(RNN) model. The context-aware NMT(RNN) model, does not improve when using 2 sentences (4a), but has large gains when extending to 3 (5a). In our context-aware models, the larger context is handled by concatenating all previous sentences. The context-aware NMT(Transformer) (4b), (5b) was actually hurt by the larger context. On the other hand, for the concatenation model (6b), (7b) we observed some improvements, but they were not as consistent as the gains for the NMT(RNN) model.

The results in (2ab), (3ab), (4ab), (5ab) (6ab), (7ab) are obtained with models trained and evaluated with the gold standard previous target sentences as context. In the fair experiments (8ab), (9ab) we train with the gold standard previous target sentence as context, but then evaluate with translations of the previous source sentences obtained with the baseline model. This lowers the performance of both NMT(RNN) models (8a), (9a), but they still improve over the baseline. Our context-aware NMT(Transformer) model (8b) slightly lowers performance compared to the baseline, unlike the concatenation model (9b).

Additionally, we train context-aware models where the previous sentence is obtained from the source side (10ab), (11ab). Even in such a scenario, context-aware and concatenation NMT(RNN) models obtain improvements over the baseline. Again, the concatenation NMT(Transformer) shows improvements over the baseline. The context-aware NMT(Transformer) was not able to make use of the source side information. Given that the encoder representations are shared this is to some extent surprising and suggests that additional encoder components are necessary to model the contextual representation.

## 6.2 Coreference

Results for coreference are also shown in Table 2. Experiments (12a) and (12b) show the results we obtained with the pronoun oracle setup. It is clear that NMT can benefit from strong coreference signals. We observed a large difference between the improvements on the development and the test set, probably because this phenomenon is not equally prominent in the datasets. In the absence of perfect CR, this setup is a reasonable proxy for obtaining coreference signals and gender information, and the context-aware models achieve large improvements over their respective baselines.

| | (a) RNN | (b) TF |
|---|---|---|
| (1) baseline | 28.57 | 29.53 |
| (2) context - gold prev. target | 29.89 | 30.13 |
| (3) concat - gold prev. target | 30.35 | 31.02 |
| (4) context - gold prev. 2 target | 29.96 | 29.57 |
| (5) context - gold prev. 3 target | 30.95 | 29.98 |
| (6) concat - gold prev. 2 target | 30.96 | 31.69 |
| (7) concat - gold prev. 3 target | 31.56 | 31.26 |
| (8) context - baseline prev. target | 29.10 | 29.25 |
| (9) concat - baseline prev. target | 29.28 | 29.89 |
| (10) context - prev. source | 29.48 | 28.80 |
| (11) concat - prev. source | 29.56 | 30.25 |
| **Coreference** | | |
| (12) context - pronoun oracle | 34.35 | 34.60 |
| (13) context - fair | 29.05 | 28.76 |
| (14) context - noisy pronoun oracle | 33.61 | 34.62 |
| (15) concat - noisy pronoun oracle | 35.59 | 35.18 |
| **Coherence** | | |
| (16) context - repeated target words | 29.83 | 29.35 |
| (17) context - repeated source words | 29.27 | 29.04 |
| (18) context - noisy rep. target words | 30.07 | 29.85 |
| (19) concat - noisy rep. target words | 30.46 | 31.25 |

Table 2: BLEU scores from all of the oracle experimental setups on the test set. Results in the first column correspond to the NMT(RNN) context-aware and concatenation models while the second column to the NMT(Transformer) ones. The number in brackets in each line is used to indicate the corresponding experiment throughout the text.

Experiments (13a) and (13b) show the results for the fair coreference setup. Using a CR tool, we identified the appropriate antecedents (to current sentence pronouns) in the previous source sentence and used them as context. The results show small improvements on the test set. This signal is significantly weaker. Moreover, only ≈ 0.3M samples had a non-empty context, meaning a pronoun was referring to a coreferent as identified by the CR tool. These results show that while weak, the context-aware NMT(RNN) model is able to utilize this signal. The NMT(Transformer) model on the other hand, was significantly hurt by this setup. We attribute this to the model not being able to handle scenarios where the majority of the samples are without context information.

In the noisy pronoun oracle setup, the context consists of the previous gold standard target sentence to which we append the target side pronouns as in the previously outlined pronoun oracle setup. The results are shown in Table 2. We can observe that the context-aware NMT(RNN) model (14a) is actually hurt by the extra information in the form of previous target sentence. We attribute the decrease to the model learning to strongly attend to all pronouns in the context. As such, in some cases, it chooses to attend to a pronoun from the previous sentence which ends up acting as noise in these models. Using oracles allowed us to easily find this important weakness in our model design. The context-aware NMT(Transformer) model (14b) is more robust to noise and had no problems identifying the appropriate information.

Using the same setting for the concatenation NMT(RNN) model (15a), achieves best performance with an absolute gain of 7.02 BLEU. Based on the obtained results in (3a), we conclude that the effects in (15a) are a compound of the capability of concatenation models to make use of the previous sentence and target side pronouns. The same effects can be observed for the NMT(Transformer) concatenation model as well (15b). However, despite the concatenation Transformer being able to obtain better results for the previous target sentence and pronoun oracle than the RNN model, the compound effect is not as strong.

### 6.3 Coherence

Table 2 shows the results we obtained for the coherence experimental setup. For the oracle setup, we identify repeated source and target words in the previous and current sentence, mark the source words and insert the target words in the context. For the fair setup, we insert repeated source words in the context. The aim with this scenario is to emphasize which words are potentially important for disambiguation. Moreover, in the oracle setup, we provide the presumably gold standard translation of the repeated word in the appropriate context.

Both scenarios (16a), (17a) obtain improvements over the baseline with the NMT(RNN) model, although not as strong as the gains with the pronoun oracle. One reason is that the number of samples with context is significantly smaller than the pronoun oracle. Another potential reason is that coherence is already modeled well by the baseline. The results indicate that obtaining coherence and disambiguating signals from past translation decisions, whether from an oracle such as in our work or from the model itself (Tu et al., 2017) is difficult. Nevertheless, the noticeable gains in BLEU we observed in our experiments confirm that further improvements can be made. The context-aware NMT(Transformer) is hurt by these oracle setups as shown in experiments (16b) and (17b) because of the lack of sufficient context.

Table 2 presents the results for the noisy coherence oracle. The context-aware NMT(RNN) model (18a) obtains improvement over the baseline of 1.5 BLEU and the concatenation model (19a) of 1.89 BLEU. This is likely a compound effect of having access to the entire previous target sentence as in (2a) and (3a) and the weak signals in the form of pointers to where disambiguation is necessary. This is to some extent matched by the Transformer experiments (18b), (19b).

### 6.4 Comparison with challenge sets

In order to assess the quality of our oracles, we also set them up on OpenSubtitles2016 En-Fr and compare them against the challenge sets proposed in Bawden et al. (2018). This allows us to compare the two methods and show whether we can draw similar conclusions about a model when evaluating it with both the oracles and challenge sets. For simplicity, we only evaluate our proposed context-aware NMT(RNN) model. We randomly sampled documents from the En-Fr dataset to create a development and test set. The challenge sets are used as provided by Bawden et al. (2018). We set up the oracles in the same way as for En-De. However, in French the pronouns *le*, *la* and *les* can also be used as definite articles. Therefore, we used MarMoT (Mueller et al., 2013) to filter out these instances.

We compare the methods by measuring the improvements a context-aware model achieves over a baseline, on our oracles and on the challenge sets. Since our oracles use target side knowledge, we use the version of the challenge sets where the previous sentence is from the target side. This provides for a fairer comparison. We train our context-aware model on the pronoun and repeated words oracle. In order to evaluate the model on the challenge sets, we train the model with the gold standard previous target sentence as context.

The baseline model obtains a score of 27.73 BLEU on the test and by design, it achieves 50% accuracy on the coreference and 50% accuracy on the coherence challenge set. Our proposed context-aware model trained on the pronoun oracle achieved 30.72 BLEU on the test set. On the repeated words oracle, it scored 28.25 BLEU. As in the En-De experimental results, our model ob-

| | |
|---|---|
| *pronoun oracle* | meine er !@#$ XPRONOUN My reading of the prophecy is that XPRONOUN it will come in 2012 |
| *reference* | Meine Textstudien ergeben, daß er 2012 kommen wird |
| *baseline* | Mein Lesen der Prophezeiung lautet, dass *es* 2012 kommen wird |
| *context* | Meine Lesung der Prophezeiung ist, dass er 2012 kommen wird |
| *repeated words oracle* | Abneigung Romulaner !@#$ If you had seen them kill your parents, you would XREP understand it is always the XREP time for those XREP feelings. |
| *reference* | Höatten Sie mit angesehen, wie Ihre Eltern getötet werden... Meine Abneigung gegen die Romulaner ist universell. |
| *baseline* | Wenn du gesehen hättest, wie sie deine Eltern töten würden, würdest du verstehen, dass es immer die Zeit für diese *Gefühle* ist. |
| *context* | Wenn du gesehen hättest, wie sie deine Eltern getötet haben, würdest du verstehen, dass es immer die Zeit für diese Abneigung ist. |
| *prev. sent. oracle* | Er dachte, die Geschichte handelte von einem Fisch. !@#$ It isn't? |
| *reference* | Tut sie nicht? |
| *baseline* | Ist *es* nicht? |
| *context* | Ist *es* nicht? |

Table 3: Samples from the qualitative analysis.

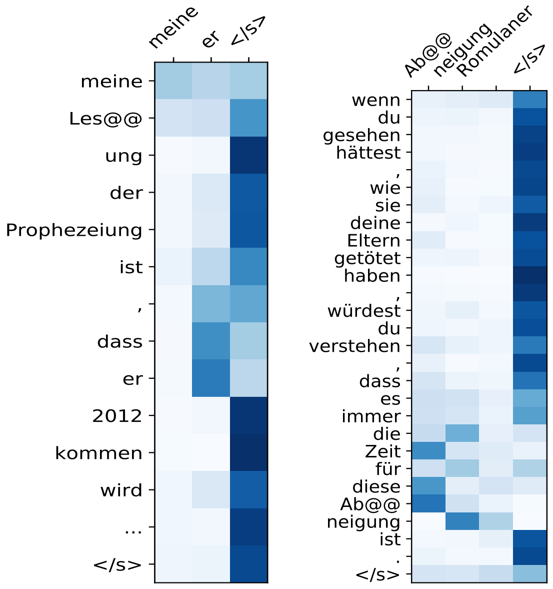
tains small gains for coherence and larger ones for coreference. The context-aware model we trained with the previous target sentence as context, scored 63.0% and 54.0%, on the coreference and coherence challenge set, respectively. From these results we also can conclude that our model is reasonably powerful to handle coreference and marginally improves coherence. These results show that challenge sets and oracles provide comparable results when evaluating discourse in MT. However, our oracle setups are easier to define and control.

### 6.5 Qualitative study

In this section, we show examples from our oracle setups and provide visualizations of the extra-sentential attention for our context-aware and the concatenation NMT(RNN) model (Tiedemann and Scherrer, 2017). We also show the activations of the decoder gates which control the context information flow. This can help us understand how the models make decisions at each time step.

In Table 3 we show the pronoun, repeated words and previous target sentence oracles and compare the output from a baseline and our proposed context-aware model against the reference translation. For simplicity, in the visualizations for the concatenation model, we only present the attention over the previous sentence and the sentence separating token SEP.

The first row in Table 3 shows a pronoun oracle sample. In this case, *it* refers to *comet*. It is obvious that there is not sufficient information in the main sentence alone to properly translate *it* and the baseline model falls back to the data-driven prior, which is to generate *es*.



(a) Pronoun (b) Repeated words

Figure 1: Context attention for the pronoun and repeated words oracles.

From the visualization in Figure 1a we see that our context-aware model pays attention to the appropriate pronoun (*meine*, *er*). From Figure 3 we see that for this example, the noisy oracle shows the same behavior and correctly ignores the noise. Furthermore, Figure 2a and Figure 2b show that the gate activations follow the intuitive assumption that they should be high when generating pronouns. Our model in the noisy pronoun oracle produced a correct translation, but it still weakly paid attention to irrelevant parts of the sentence. From Figure 4 we see that concatenation model on the other hand, makes a clean distinction between what is relevant and what is not, and only has strong attention over the pronouns.

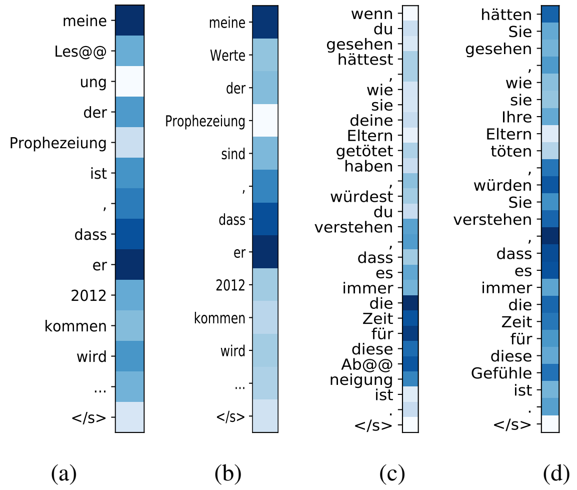


Figure 2: Gate activations for pronoun and repeated words oracles. (a) pronoun oracle, (b) - noisy pronoun oracle, (c) - repeated words oracle, (d) - noisy repeated words oracle.

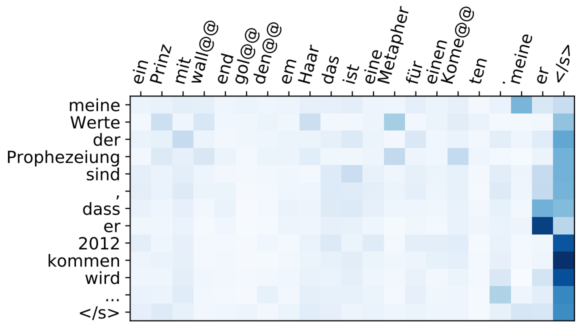


Figure 3: Context attention of our proposed model on the noisy pronoun oracle.

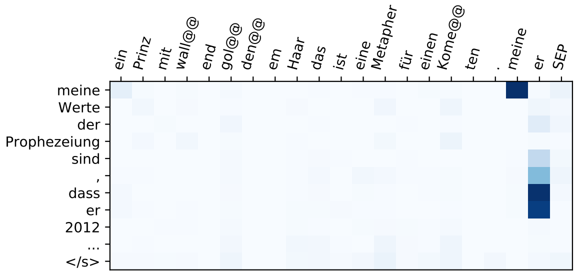


Figure 4: Attention over the previous sentence of the concatenation model on the noisy pronoun oracle.

The second sample is selected from the repeated words oracle setup. Because the reference translation does not exactly match the source sentence, there is a small mismatch between the repeated words on the source and target side. However, we see that without the contextual signal that *feelings* in this case refers to *adverse feelings* (as indicated by *Abneigung*) the baseline falls back to the more common translation *Gefühle*. We also looked at the previous sentence which did not have any context information and both the baseline and the context-aware model generated *Gefühle*.

Figure 1b shows that the context-aware model has no problem attending to the disambiguating signal (*Abneigung*) and it also uses this signal when generating the determiner *dieses* which is dependent on the noun. However, we also can observe that given the incorrect indication to look at the context when translating *time*, it also has attention activation over the context as well. This is closely followed by the gate activations in Figure 2c. The same doesn't happen when translating the marked source token *understand*. This is probably because the model is confident that it doesn't need context when translating *understand*.

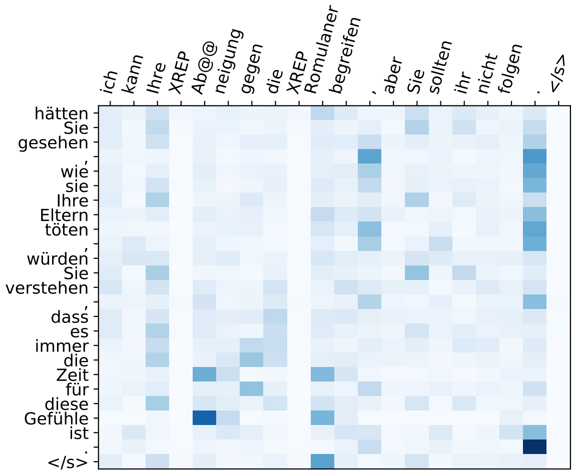


Figure 5: Context attention of our proposed model on the noisy repeated words oracle.

From Figure 5 and Figure 2d we see that the context-aware model in a noisy repeated words oracle setting has difficulties identifying the coherence information and when to use it. It tends to pay attention to certain words throughout the whole sequence generation. This is likely a side effect of having access to the previous target sentence which in other cases provides useful information. Although it pays attention to the appropriate repeated word (*Abneigung*), it still fails to generate it. Since the concatenation model uses an RNN over the context, it has no problem identifying the disambiguating signal, marked with XREP and generates it accordingly (Figure 6).

We also did an analysis of the previous target sentence oracle as well as the models that use the previous source sentence as context. We looked at examples where there is an anaphoric pronoun *it*. When the context is from the source side, our

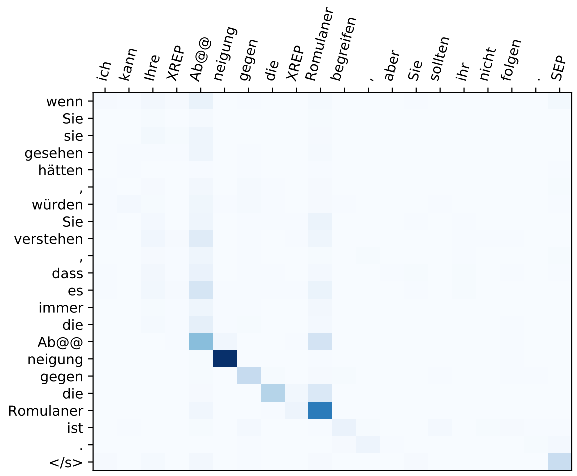


Figure 6: Attention over the previous sentence of the concatenation model on the noisy repeated words oracle.

context-aware model tends to pay attention to a single noun, while in the previous target sentence oracle, it looks at more explicit gender information, such as pronouns, articles etc. This is illustrated in the last example in Table 3 and Figure 7 and 8. In this case, *it* refers to *die Geschichte* or *story*. When translating *it* both models paid attention to the appropriate place in the previous sentence, but failed to generate the correct pronoun *sie*. For this particular example, the concatenation model paid no attention to the previous sentence.

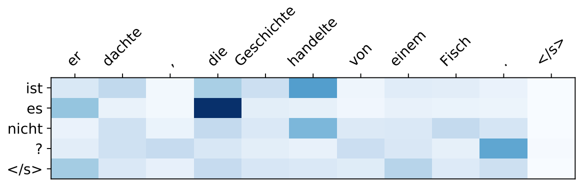


Figure 7: Context attention of our proposed model on the previous target sentence.

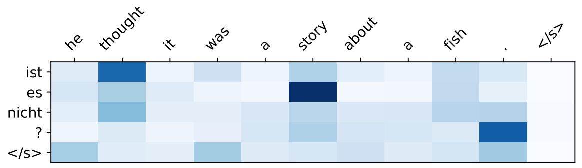


Figure 8: Context attention of our proposed model on the previous source sentence.

### 6.6 Model inference speed

Although the concatenation model performs better than our context-aware model, an important consideration when working with context-aware NMT is computational efficiency. We compared inference times for the RNN models on the development set. We report times with context size of 1, 2 and 3 previous sentences.

The context model took 1233 seconds to decode the development set, while the concatenation model 2063 seconds. The concatenation model took additional $\approx$ 900 seconds for each additional context sentence. Because our context-aware implementation is not tightly dependent on context length, there are no considerable drops in speed. This is a disadvantage of the concatenation approach. If one is to use large context, or even entire documents, the problem quickly becomes very computationally expensive. This highlights the necessity of specialized context-aware models. Since the Transformer can be more easily parallelized, there is still room for improving the computational performance of our context-aware Transformer. As a result, we leave such a comparison for future work.

## 7 Conclusion and Future Work

We used simple oracles to look at discourse-level phenomena in MT. We compared context-aware NMT models and show that these approaches provide large gains in BLEU for coreference and coherence given clear oracle signals. We also showed that even when using fair signals, such as the previous source sentence or a system translation of the previous target sentence, NMT models benefit and make use of the extra information. Some future work in context-aware NMT can focus on using the standard NMT architecture, which performs well. However, if one requires access to larger context, vanilla NMT will have difficulties scaling in terms of speed and perhaps even in modeling ability. For this reason, a promising way forward is studying different ways of modeling and integrating context that support fast inference. Oracle experiments will allow us to quickly test interesting modeling differences.

## Acknowledgments

We would like to thank the anonymous reviewers for their valuable input and Daniel Ledda for his help with examples. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement № 640550).

## References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, ICLR '15. ArXiv: 1409.0473.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating Discourse Phenomena in Neural Machine Translation. In *NAACL 2018*, New Orleans, USA.

Marine Carpuat. 2009. One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 19–27.

Marine Carpuat and Michel Simard. 2012. The trouble with smt consistency. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 442–449.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.

Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19.

Liane Kirsten Guillou. 2016. *Incorporating pronoun function into statistical machine translation*. Ph.D. thesis, The University of Edinburgh, UK.

Christian Hardmeier. 2012. Discourse in statistical machine translation. a survey and a case study. *Discours. Revue de linguistique, psycholinguistique et informatique. A journal of linguistics, psycholinguistics and computational linguistics*, (11).

Christian Hardmeier and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In *IWSLT (International Workshop on Spoken Language Translation); Paris, France; December 2nd and 3rd, 2010.*, pages 283–289.

Christian Hardmeier, Sara Stymne, Jörg Tiedemann, and Joakim Nivre. 2013. Docent: A document-level decoder for phrase-based statistical machine translation. In *ACL 2013 (51st Annual Meeting of the Association for Computational Linguistics); 4-9 August 2013; Sofia, Bulgaria*, pages 193–198.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Yangfeng Ji, Trevor Cohn, Lingpeng Kong, Chris Dyer, and Jacob Eisenstein. 2015. Document context language models. *arXiv preprint arXiv:1511.03962*.

Ronan Le Nagard and Philipp Koehn. 2010. Aiding pronoun translation with co-reference resolution. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 252–261.

Sharid Loáiciga, Sara Stymne, Preslav Nakov, Christian Hardmeier, Jörg Tiedemann, Mauro Cettolo, and Yannick Versley. 2017. Findings of the 2017 discomt shared task on cross-lingual pronoun prediction. In *The Third Workshop on Discourse in Machine Translation*.

Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163.

Tim Salimans and Diederik P Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016.

Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2017. Learning to remember translation history with a continuous cache. *arXiv preprint arXiv:1711.09367*.

Ferhan Ture, Douglas W Oard, and Philip Resnik. 2012. Encouraging consistent translation choices. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 417–426.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-Aware Neural Machine Translation Learns Anaphora Resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831.

Tian Wang and Kyunghyun Cho. 2016. Larger-context language modelling with recurrent neural network. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, volume 3, pages 1319–1329.

# A Large-Scale Test Set for the Evaluation of Context-Aware Pronoun Translation in Neural Machine Translation

**Mathias Müller**[1,2] **Annette Rios**[1] **Elena Voita**[3,4] **Rico Sennrich**[1,5]

[1]Institute of Computational Linguistics, University of Zurich
[2]Amazon Research, Berlin*
[3]Yandex Research, Russia [4]University of Amsterdam, Netherlands
[5]School of Informatics, University of Edinburgh

## Abstract

The translation of pronouns presents a special challenge to machine translation to this day, since it often requires context outside the current sentence. Recent work on models that have access to information across sentence boundaries has seen only moderate improvements in terms of automatic evaluation metrics such as BLEU. However, metrics that quantify the overall translation quality are ill-equipped to measure gains from additional context. We argue that a different kind of evaluation is needed to assess how well models translate inter-sentential phenomena such as pronouns. This paper therefore presents a test suite of contrastive translations focused specifically on the translation of pronouns. Furthermore, we perform experiments with several context-aware models. We show that, while gains in BLEU are moderate for those systems, they outperform baselines by a large margin in terms of accuracy on our contrastive test set. Our experiments also show the effectiveness of parameter tying for multi-encoder architectures.

## 1 Introduction

Even though machine translation has improved considerably with the advent of neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015), the translation of pronouns remains a major issue. They are notoriously hard to translate since they often require context outside the current sentence.

**EN** However, the European Central Bank (ECB) took an interest in it. *It* describes bitcoin as "the most successful virtual currency".

**DE** Dennoch hat die Europäische Zentralbank (EZB) Interesse hierfür gezeigt. *Sie* beschreibt Bitcoin als "die virtuelle Währung mit dem grössten Erfolg".

Figure 1: Example sentence illustrating how the translation of pronouns is ambiguous on a sentence level. Pronouns of interest are in italics, and the antecedents they refer to are underlined. Taken from WMT `newstest2013`.

As an example, consider the sentences in Figure 1. In both languages, there is a pronoun in the second sentence that refers to the European Central Bank. When the second sentence is translated from English to German, the translation of the pronoun *it* is ambiguous. This ambiguity can only be resolved with context awareness: if a translation system has access to the previous English sentence, the previous German translation, or both, it can determine the antecedent the pronoun refers to. In this German sentence, the antecedent *Europäische Zentralbank* dictates the feminine gender of the pronoun *sie*.

It is unfortunate, then, that current NMT systems generally operate on the sentence level (Vaswani et al., 2017; Gehring et al., 2017; Hieber et al., 2017). Documents are translated sentence-by-sentence for practical reasons, such as line-based processing in a pipeline and reduced computational complexity. Furthermore, improvements of larger-context models over baselines in terms of document-level metrics such as BLEU or RIBES have been moderate, so that their computational overhead does not seem justified, and so that it is hard to develop more effective context-aware architectures and empirically validate them.

[*] Work performed prior to joining Amazon.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 61–72
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64007

To address this issue, we present an alternative way of evaluating larger-context models on a test set that allows to specifically measure a model's capability to correctly translate pronouns. The test suite consists of pairs of source and target sentences, in combination with contrastive translation variants (for evaluation by model scoring) and additional linguistic and contextual information (for further analysis). The resource is freely available.[1] Additionally, we evaluate several context-aware models that have recently been proposed in the literature on this test set, and extend existing models with parameter tying.

The main contributions of our paper are:

- We present a large-scale test set to evaluate the accuracy with which NMT models translate the English pronoun *it* to its German counterparts *es*, *sie* and *er*.
- We evaluate several context-aware systems and show how targeted, contrastive evaluation is an effective tool to measure improvement in pronoun translation.
- We empirically demonstrate the effectiveness of parameter tying in multi-encoder context-aware models.

Section 2 explains how our paper relates to existing work on context-aware models and the evaluation of pronoun translation. Section 3 describes our test suite. The context-aware models we use in our experiments are detailed in Section 4. We discuss our experiments in Section 5 and the results in Section 6.

## 2 Related Work

Two lines of work are related to our paper: research on context-aware translation (described in Section 2.1) and research on focused evaluation of pronoun translation (described in Section 2.2).

### 2.1 Context-Aware NMT Models

If the translation of a pronoun requires context beyond the current sentence (see the example in Figure 1), a natural extension of sentence-level NMT models is to condition the model prediction on this necessary context. In the following, we describe a number of existing approaches to making models "aware" of additional context.

The simplest possible extension is to translate units larger than sentences. Tiedemann and Scherrer (2017) concatenate each sentence with the sentence that precedes it, for the source side of the corpus or both sides. All of their models are standard sequence-to-sequence models built with recurrent neural networks (RNNs), since the method does not require any architectural change. Agrawal et al. (2018) use the same concatenation technique with a Transformer architecture (Vaswani et al., 2017), and experiment with wider context.

A number of works do propose changes to the NMT architecture. A common technique is to extend a standard encoder-decoder model by additional encoders for the context sentence(s), with a modified attention mechanism (Jean et al., 2017; Bawden et al., 2018; Voita et al., 2018). One aspect that differs between these works is the architecture of the encoder and attention. While Jean et al. (2017); Bawden et al. (2018) extend an RNN encoder-decoder with a second encoder that the decoder attends to, Voita et al. (2018) extend the Transformer architecture with an encoder that is attended to by the main encoder. Voita et al. (2018) also introduce parameter sharing between the main encoder and the context encoder, but do not empirically demonstrate its importance.

While the number of encoded sentences in the previous work is fixed, Wang et al. (2017); Maruf and Haffari (2018) explore the integration of variable-size context through a hierarchical architecture, where a first-level RNN reads in words to produce sentence vectors, which are then fed into a second-level RNN to produce a document summary.

Apart from differences in the architectures, related work varies in whether it considers source context, target context, or both (see Table 1 for an overview of language arcs and context types). Some work considers only source context, but for pronoun translation, target-side context is intuitively important for disambiguation, especially if the antecedent itself is ambiguous. In our evaluation, we therefore emphasize models that take into account both source and target context.

Our experiments are based on models from Bawden et al. (2018), who have released their source code.[2] We extend their models with parameter sharing, which was shown to be beneficial

[1] https://github.com/ZurichNLP/ContraPro

[2] https://github.com/rbawden/nematus

| | Languages | | Context types | | | |
|---|---|---|---|---|---|---|
| | source | target | source | target | preceding | following |
| Tiedemann and Scherrer (2017) | DE | EN | x | x | x | |
| Jean et al. (2017) | EN | FR/DE | x | | x | |
| Wang et al. (2017) | ZH | EN | x | | x | |
| Voita et al. (2018) | EN | RU | x | | x | x |
| Bawden et al. (2018) | EN | FR | x | x | x | |
| Maruf and Haffari (2018) | FR/DE/ET | EN | x | x | x | |
| Agrawal et al. (2018) | EN | IT | x | x | x | x |

Table 1: Overview of context-aware translation models in related work.

by Voita et al. (2018). Additionally, we consider a concatenative baseline, similar to Tiedemann and Scherrer (2017), and Transformer-based models (Voita et al., 2018).

### 2.2 Evaluation of Pronoun Translation

Pronouns can serve a variety of functions with complex cross-lingual variation (Guillou, 2016), and hand-picked, manually annotated test suites have been presented for the evaluation of pronoun translation (Guillou and Hardmeier, 2016; Isabelle et al., 2017; Bawden et al., 2018). While suitable for analysis, the small size of the test suites makes it hard to make statistically confident comparisons between systems, and the hand-picked nature of the test suites introduces biases.[3] To overcome these problems, we opted for a fully automatic approach to constructing a large-scale test suite.

Conceptually, our test set is most similar to the "cross-lingual pronoun prediction" task held at DiscoMT and WMT in recent years (Hardmeier et al., 2015; Guillou et al., 2016; Loáiciga et al., 2017): participants are asked to fill a gap in a target sentence, where gaps correspond to pronouns.

The first edition of the task focused on English→French, and it was found that local context (such as the verb group) was a strong signal for pronoun prediction. Hence, future editions only provided target-side lemmas instead of fully inflected forms, which makes the task less suitable to evaluate end-to-end neural machine translation systems, although such systems have been trained on the task (Jean et al., 2017).

Loáiciga et al. (2017) do not report on the proportion of intra-sentential and inter-sentential anaphora in their test set, but the two top-performing systems only made use of intra-sentential information. Our test suite focuses on allowing the comparison of end-to-end context-aware NMT systems, and we thus extract a large number of *inter-sentential anaphora*, with meta-data allowing for a focus on inter-sentential anaphora with a long distance between the pronoun and its antecedent. Our focus on evaluating end-to-end NMT systems also relieves us from having to provide annotated training sets, and reduces pressure to achieve balance and full coverage of phenomena.[4]

An alternative approach to automatically evaluate pronoun translation are reference-based methods that produce a score based on word alignment between source, translation output, and reference translation, and identification of pronouns in them, such as AutoPRF (Hardmeier and Federico, 2010) and APT (Miculicich Werlen and Popescu-Belis, 2017). Guillou and Hardmeier (2018) perform a human meta-evaluation and show substantial disagreement between reference-based metrics and human judges, especially because there often exist valid alternative translations that use different pronouns than the reference. Our test set, and our protocol of generating contrastive examples, is focused on selected pronouns to minimize the risk of producing contrastive examples that are actually valid translations.

## 3 Test set with contrastive examples

Contrastive evaluation requires a large set of suitable examples that involve the translation of pronouns. As additional goals, our test set is designed

[3] For example, all pronoun examples in the test suite by Bawden et al. (2018) require the previous target sentence for disambiguation, and thus do not reward models that condition on more than one sentence of context.

[4] For example, we do not consider cases where English *it* is translated into something other than a personal pronoun. While this would be a severe blind spot in a training set for pronoun prediction, the focused nature of our test suite does not impair the performance of end-to-end NMT systems on other phenomena.

| Alignment | Frequency | Probability |
|---|---:|---:|
| it→es | 255764 | 0.334 |
| it→sie | 64446 | 0.084 |
| it→er | 44543 | 0.058 |
| it→ist | 42614 | 0.055 |
| it→Sie | 26054 | 0.034 |
| it→, | 21037 | 0.027 |
| it→das | 17992 | 0.023 |
| it→dies | 11943 | 0.015 |
| it→wird | 11886 | 0.015 |
| it→man | 10539 | 0.013 |
| it→ihn | 7744 | 0.010 |

Table 2: Frequency and probability of alignments of *it* in the training data of our systems (all data from the WMT 2017 news translation task). Alignments are produced by a fast_align model.

to 1) focus on *hard* cases, so that it can be used as a benchmark to track progress in context-aware translation and 2) allow for fine-grained analysis.

Section 3.1 describes how we extract our data set. Section 3.2 explains how, given a set of contrastive examples, contrastive evaluation works.

### 3.1 Automatic extraction of contrastive examples from corpora

We automatically create a test set from the OpenSubtitles corpus (Lison and Tiedemann, 2016).[5] The goal is to provide a large number of difficult test cases where an English pronoun has to be translated to a German pronoun.

The most challenging cases are translating *it* to either *er, sie* or *es*, depending on the grammatical gender of the antecedent.[6] Not only is the translation of *it* ambiguous, there is also class imbalance in the training data (see Table 2). There is roughly a 30% probability that *it* is aligned to *es*,[7] which makes it difficult to learn to translate *er* and *sie*. We use parsing and automatic co-reference resolution to find translation pairs that satisfy our constraints.

To provide a basis for filtering with constraints, we tokenize the whole data set with the Moses tokenizer, generate symmetric word alignments with fast_align (Dyer et al., 2013), parse the English text with CoreNLP (Manning et al., 2014), parse the German text with ParZu (Sennrich et al., 2013) and perform coreference resolution on both sides. The coreference chains are obtained with the neural model of CoreNLP for English, and with CorZu for German (Tuggener, 2016), respectively.

Then we opt for high-precision, aggressive filtering, according to the following protocol: for each pair of sentences $(e, f)$ in English and German, extract iff

- $e$ contains the English pronoun *it*, and $f$ contains a German pronoun that is third person singular (*er, sie* or *es*), as indicated by their part-of-speech tags;
- those pronouns are aligned to each other;
- both pronouns are in a coreference chain;
- their nominal antecedents in the coreference chain are aligned on word level.

This removes most candidate pairs, but is necessary to overcome the noise introduced by our preprocessing pipeline, most notably coreference resolution. From the filtered set, we create a balanced test set by randomly sampling 4000 instances of each of the three translations of *it* under consideration (*er*, *sie*, *es*). We do not balance antecedent distance. See Table 4 for the distribution of pronoun pairs and antecedent distance in the test set.

For each sentence pair in the resulting test set, we introduce *contrastive translations*. A contrastive translation is a translation variant where the correct pronoun is swapped with an incorrect one. For an example, see Table 3, where the pronoun *it* in the original translation corresponds to *sie* because the antecedent *bat* is a feminine noun in German (*Fledermaus*). We produce wrong translations by replacing *sie* with one of the other pronouns (*er*, *es*).

Note that, by themselves, these contrastive translations are grammatically correct if the antecedent is outside the current sentence. The test set also contains pronouns with an antecedent in the same sentence (antecedent distance 0). Those examples do not require any additional context

[5] http://opus.nlpl.eu/OpenSubtitles2016.php

[6] The pronouns *he* and *she* usually refer to a person in English, and since persons do not change gender in the translation, we assume that learning the correspondences *he* → *er* and *she* → *sie* does not present a challenge for a model. Cases where *he* or *she* refer to a noun that is not a person are possible, but extremely rare.

[7] Note that these statistics include non-referential uses of *it*, that we exclude from our testset.

| | |
|---|---|
| source: | *It could get tangled in your hair.* |
| reference: | ***Sie** könnte sich in deinem Haar verfangen.* |
| contrastive: | ***Er** könnte sich in deinem Haar verfangen.* |
| contrastive: | ***Es** könnte sich in deinem Haar verfangen.* |
| antecedent en: | a bat |
| antecedent de: | eine Fledermaus (f.) |
| antecedent distance : | 1 |

Table 3: Example sentence pair with contrastive translations. An antecedent distance of 1 means that the antecedent is in the immediately preceding sentence.

for disambiguation and we therefore expect the sentence-level baseline to perform well on them.

We take extra care to ensure that the resulting contrastive translations are grammatically correct, because ungrammatical sentences are easily dismissed by an NMT system. For instance, if there are any possessive pronouns (such as *seine*) in the sentence, we also change their gender to match the personal pronoun replacement.

The German coreference resolution system does not resolve *es* because most instances of *es* in German are either non-referential forms, or they refer to a clause instead of a nominal antecedent. We limit the test set to nominal antecedents, as these are the only ambiguous cases with respect to translation. For this reason, we have to rely entirely on the English coreference links for the extraction of sentence pairs with *it→es*, as opposed to pairs with *it→er* and *it→sie* where we have coreference chains in both languages.[8]

Our extraction process respects document boundaries, to ensure we always search for the right context. We extract additional information from the annotated documents, such as the distance (in sentences) between pronouns and their antecedents, the document of origin, lemma, morphology and dependency information if available.

### 3.2 Evaluation by scoring

Contrastive evaluation is different from conventional evaluation of machine translation in that it does not require any translation. Rather than testing a model's ability to translate, it is a method to test a model's ability to *discriminate* between given good and bad translations.

| distance | *it→es* | *it→er* | *it→sie* | total |
|---|---|---|---|---|
| 0 | 872 | 736 | 792 | 2400 |
| 1 | 1892 | 2577 | 2606 | 7075 |
| 2 | 631 | 459 | 420 | 1510 |
| 3 | 274 | 167 | 132 | 573 |
| >3 | 331 | 61 | 50 | 442 |
| total | 4000 | 4000 | 4000 | 12000 |

Table 4: Test set frequencies of pronoun pairs and antecedent distance (measured in sentences).

We exploit the fact that NMT systems are in fact language models of the target language, conditioned on source text. Like language models, NMT systems can be used to compute a model score (the negative log probability) for an existing translation. Contrastive evaluation, then, means to compare the model score of two pairs of inputs: $(actual\ source,\ reference\ translation)$ and $(actual\ source,\ contrastive\ translation)$. If the model score of the actual reference translation is higher, we assume that this model can detect wrong pronoun translations.

However, this does *not* mean that systems actually produce the reference translation when given the source sentence for translation. An entirely different target sequence might rank higher in the system's beam during decoding. The only conclusion permitted by contrastive evaluation is whether or not the reference translation is more probable than a contrastive variant.

If the model score of the reference is indeed higher, we refer to this outcome as a "correct decision" by the model. The model's decision is only correct if the reference translation has a higher score than any contrastive translation. In our evaluation, we aggregate model decisions on

[8] There are some cases where the antecedent is listed as *it* in the test set. This is our fallback behaviour if the coreference chain does not contain any noun. In that case, we do not know the true antecedent.

the whole test set and report the overall percentage of correct decisions as accuracy.

During scoring, the model is provided with reference translations as target context, while during translation, the model needs to predict the full sequence. It is an open question to what extent performance deteriorates when context is itself predicted, and thus noisy. We highlight that the same problem arises for sentence-level NMT, and has been addressed with alternative training strategies (Ranzato et al., 2015).

## 4 Context-Aware NMT Models

This section describes several context-aware NMT models that we use in our experiments. They fall into two major categories: models based on RNNs and models based on the Transformer architecture (Vaswani et al., 2017). We experiment with additional context on the source side and target side.

### 4.1 Recurrent Models

We consider the following recurrent baselines:

**baseline** Our baseline model is a standard bidirectional RNN model with attention, trained with Nematus. It operates on the sentence level and does not see any additional context. The input and output embeddings of the decoder are tied, encoder embeddings are not.

**concat22** We concatenate each sentence with one preceding sentence, for both the source and target side of the corpus. Then we train on this new data set without any changes to the model architecture. This very simple method is inspired by Tiedemann and Scherrer (2017).

The following models are taken, or slightly adapted, from Bawden et al. (2018). For this reason, we give only a very short description of them here and the reader is referred to their work for details.

**s-hier** A multi-encoder architecture with hierarchical attention. This model has access to one additional context: the previous source sentence. It is read by a separate encoder, and attended to by an additional attention network. The output of the resulting two attention vectors is combined with yet another attention network.

**s-t-hier** Identical to *s-hier*, except that it considers two additional contexts: the previous source sentence and previous target sentence. Both are read by separate encoders, and sequences from all encoders are combined with hierarchical attention.

**s-hier-to-2** The model has an additional encoder for source context, whereas the target side of the corpus is concatenated, in the same way as for *concat22*. This model achieved the best results in Bawden et al. (2018).

For each variant, we also introduce and test weight tying: we share the parameters of embedding matrices between encoders that read the same kind of text (source or target side).

### 4.2 Transformer Models

All remaining models are based on the Transformer architecture (Vaswani et al., 2017). A Transformer avoids recurrence completely: it follows an encoder-decoder architecture using stacked self-attention and fully connected layers for both the encoder and decoder.

**baseline** A standard context-agnostic Transformer. All model parameters are identical to a *Transformer-base* in Vaswani et al. (2017).

**concat22** A simple concatentation model where only the training data is modified, in the same way as for the recurrent *concat22* model.

**concat21** Trained on data where the preceding sentence is concatenated to the current one only on the source side. This model is also taken from Tiedemann and Scherrer (2017).

**Voita et al. (2018)** A more sophisticated context-aware Transformer that uses source context only. It has a separate encoder for source context, but all layers except the last one are shared between encoders. A source and context sentence are first encoded independently, and then a single attention layer and a gating function are used to produce a context-aware representation of the source sentence. Such restricted interaction with context is shown to be beneficial for analysis of contextual phenomena captured by the model. For details the reader is referred to their work.

## 5 Experiments

We train all models on the data from the WMT 2017 English→German news translation shared task ($\sim$ 5.8 million sentence pairs). These corpora do not have document boundaries, therefore a small fraction of sentences will be paired with wrong context, but we expect the model to be robust against occasional random context (see also Voita et al. 2018). Experimental setups for the RNN and Transformer models are different, and we describe them separately.

All RNN-based models are trained with Nematus (Sennrich et al., 2017). We learn a joint BPE model with 89.5k merge operations (Sennrich et al., 2016). We train shallow models with an embedding size of 512, a hidden layer size of 1024 and layer normalization. Models are trained with Adam (Kingma and Ba, 2015), with an initial learning rate of 0.0001. We apply early stopping based on validation perplexity. The batch size for training is 80, and the maximum length of training sequences is 100 (if input sentences are concatenated) or 50 (if input lines are single sentences).

For our Transformer-based experiments, we use a custom implementation and follow the hyperparameters from Vaswani et al. (2017); Voita et al. (2018). Systems are trained on lowercased text that was encoded using BPE (32k merge operations). Models consist of 6 encoder and decoder layers with 8 attention heads. The hidden state size is 512, the size of feedforward layers is 2048.

Model performance is evaluated in terms of BLEU, on `newstest2017`, `newstest2018` and all sentence pairs from our pronoun test set. We compute scores with SacreBLEU (Post, 2018).[9] Evaluation with BLEU is done mainly to control for overall translation quality.

To evaluate pronoun translation, we perform contrastive evaluation and report the accuracy of models on our contrastive test set.

## 6 Evaluation

The BLEU scores in Table 5 show a moderate improvement for most context-aware systems. This suggests that the architectural changes for the context-aware models do not degrade overall translation quality. The contrastive evaluation on our test set on the other hand shows a clear increase in the accuracy of pronoun translation: The best model *s-hier-to-2.tied* achieves a total of +16 percentage points accuracy on the test set over the baseline, see Table 6.

Table 7 shows that context-aware models perform better than the baseline when the antecedent is outside the current sentence. In our experiments, all context-aware models consider one preceding sentence as context. The evaluation according to the distance of the antecedent in Table 8 confirms that the subset of sentences with antecedent distance 1 benefits most from the tested context-aware models (up to +20 percentage points accuracy). However, we note two surprising patterns:

- For inter-sentential anaphora, the performance of all systems, including the baseline, improves with increasing antecedent distance.
- Context-aware systems that consider one preceding sentence also improve on intra-sentential anaphora, and on pronouns whose antecedent is outside the context window.

The first observation can be explained by the distribution of German pronouns in the test set. The further away the antecedent, the higher the percentage of *it*→*es* cases, which are the majority class, and thus the class that will be predicted most often if evidence for other classes is lacking. We speculate that this is due to our more permissive extraction heuristics for *it*→*es*.

We attribute the second observation to the existence of coreference chains where the preceding sentence contains a pronoun that refers to the same nominal antecedent as the pronoun in the current sentence. Consider the example in Table 9: The nominal antecedent of *it* in the current sentence is *door*, *Tür* in German with feminine gender. The nominal antecedent occurs two sentences before the current sentence, but the German sentence in between contains the pronoun *sie*, which is a useful signal for the context-aware models, even though they cannot know the nominal antecedent.

Note that only models aware of target-side context can benefit from such circumstances: The *s-hier* models as well as the Transformer model by (Voita et al., 2018) only see source side context, which results in lower accuracy if the distance to the antecedent is >1, see Table 8.

While such coreference chains complicate the interpretation of the results, we note that improvements on inter-sentential anaphora with antecedent distance $>$ 1 are relatively small (compared to distance 1), and that performance is still relatively poor (especially for the minority classes *er* and *sie*). We encourage evaluation of wider-context models on this subset, which is still large thanks to the size of the full test set.

Regarding the comparison of different context-aware architectures, our results demonstrate the

[9] Our (cased) SacreBLEU signature is `BLEU+c.mixed+l.en-de+#.1+s.exp+t.wmt{17,18}+tok.13a+v.1.2.10`.

| | newstest2017 | | newstest2018 | | pronoun set | |
|---|---|---|---|---|---|---|
| | cased | uncased | cased | uncased | cased | uncased |
| baseline | 23.0 | 23.7 | 33.7 | 34.2 | 19.4 | 19.9 |
| concat22 | 23.8 | 24.4 | **34.5** | 35.0 | **20.2** | 20.8 |
| **independent encoders** | | | | | | |
| s-hier | 23.5 | 24.0 | 33.5 | 34.0 | 18.9 | 19.5 |
| s-hier-to-2 | 23.8 | 24.3 | 34.2 | 34.8 | 19.2 | 19.7 |
| s-t-hier | 23.1 | 23.6 | 33.1 | 33.6 | 19.3 | 20.0 |
| **with weight tying** | | | | | | |
| s-hier.tied | 23.6 | 24.1 | 33.7 | 34.2 | 19.7 | 20.3 |
| s-hier-to-2.tied | **24.2** | 24.8 | 34.1 | 34.7 | 20.1 | 20.7 |
| s-t-hier.tied | 23.5 | 24.0 | 33.9 | 34.5 | 19.4 | 20.0 |
| **Transformer-based models** | | | | | | |
| baseline | - | 24.6 | - | 35.4 | - | 21.1 |
| concat21 | - | 24.8 | - | 35.3 | - | **21.8** |
| concat22 | - | 24.4 | - | 36.0 | - | 21.3 |
| (Voita et al., 2018) | - | **25.3** | - | **36.5** | - | 21.7 |

Table 5: English→German BLEU scores on newstest2017, newstest2018 and all sentence pairs from our pronoun test set. Case-sensitive and case-insensitive (uncased) scores are reported. Higher is better, and the best scores are marked in bold.

| | | reference pronoun | | |
|---|---|---|---|---|
| | total | *es* | *er* | *sie* |
| baseline | 0.44 | 0.85 | 0.17 | 0.31 |
| concat22 | 0.53 | 0.84 | 0.32 | 0.42 |
| **independent encoders** | | | | |
| s-hier | 0.43 | 0.80 | 0.20 | 0.29 |
| s-hier-to-2 | 0.55 | 0.84 | 0.41 | 0.40 |
| s-t-hier | 0.52 | 0.88 | 0.32 | 0.36 |
| **with weight tying** | | | | |
| s-hier.tied | 0.47 | 0.85 | 0.30 | 0.26 |
| s-hier-to-2.tied | **0.60** | 0.87 | **0.45** | **0.48** |
| s-t-hier.tied | 0.56 | 0.86 | 0.39 | 0.42 |
| **Transformer-based models** | | | | |
| baseline | 0.47 | 0.81 | 0.22 | 0.38 |
| concat21 | 0.48 | 0.88 | 0.26 | 0.31 |
| concat22 | 0.49 | **0.91** | 0.20 | 0.36 |
| (Voita et al., 2018) | 0.49 | 0.84 | 0.23 | 0.39 |

Table 6: Accuracy on contrastive test set (N=4000 per pronoun) with regard to reference pronoun.

| | antecedent location | |
|---|---|---|
| | intrasegmental | external |
| baseline | 0.57 | 0.41 |
| concat22 | 0.58 | 0.51 |
| **independent encoders** | | |
| s-hier | 0.58 | 0.39 |
| s-hier-to-2 | 0.63 | 0.53 |
| s-t-hier | 0.52 | 0.52 |
| **with weight tying** | | |
| s-hier.tied | 0.56 | 0.45 |
| s-hier-to-2.tied | 0.65 | **0.58** |
| s-t-hier.tied | 0.57 | 0.55 |
| **Transformer-based models** | | |
| baseline | 0.70 | 0.41 |
| concat21 | 0.67 | 0.44 |
| concat22 | 0.56 | 0.47 |
| (Voita et al., 2018) | **0.75** | 0.43 |

Table 7: Accuracy on contrastive test set with regard to antecedent location (within segment vs. outside segment).

| | antecedent distance | | | | |
|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | >3 |
| baseline | 0.57 | 0.38 | 0.47 | 0.52 | 0.67 |
| concat22 | 0.58 | 0.50 | 0.51 | 0.51 | 0.69 |
| **independent encoders** | | | | | |
| s-hier | 0.58 | 0.36 | 0.42 | 0.46 | 0.61 |
| s-hier-to-2 | 0.63 | 0.51 | 0.54 | 0.60 | 0.70 |
| s-t-hier | 0.52 | 0.49 | **0.57** | **0.61** | 0.71 |
| **with weight tying** | | | | | |
| s-hier.tied | 0.56 | 0.43 | 0.46 | 0.49 | 0.67 |
| s-hier-to-2.tied | 0.65 | **0.58** | 0.55 | 0.55 | **0.75** |
| s-t-hier.tied | 0.57 | 0.54 | 0.56 | 0.59 | 0.72 |
| **Transformer-based models** | | | | | |
| baseline | 0.70 | 0.38 | 0.45 | 0.49 | 0.65 |
| concat21 | 0.67 | 0.42 | 0.45 | 0.47 | 0.66 |
| concat22 | 0.56 | 0.44 | 0.53 | 0.54 | 0.74 |
| (Voita et al., 2018) | **0.75** | 0.39 | 0.48 | 0.54 | 0.66 |

Table 8: Accuracy on contrastive test set with regard to antecedent distance of antecedent (in sentences).

| | |
|---|---|
| source sentence with antecedent | *What's with the door?* |
| target sentence with antecedent | *Was ist mit der Tür?* |
| source context | ***It** won't open.* |
| reference context | ***Sie** geht nicht auf.* |
| source sentence | *- Is **it** locked?* |
| reference sentence | *- Ist **sie** abgeschlossen?* |
| contrastive 1 | *- Ist **er** abgeschlossen?* |
| contrastive 2 | *- Ist **es** abgeschlossen?* |

Table 9: Example where 1) antecedent distance is >1 and 2) the context given contains another pronoun as an additional hint.

effectiveness of parameter sharing between the main encoder (or decoder) and the contextual encoder. We observe an improvement of 5 percentage points from *s-hier-to-2* to *s-hier-to-2.tied*, and 4 percentage points from *s-t-hier* to *s-t-hier.tied*. Context encoders introduce a large number of extra parameters, while inter-sentential context is only relevant for a relatively small number of predictions. We hypothesize that the training signal is thus too weak to train a strong contextual encoder in an end-to-end fashion without parameter sharing. Our results also confirm the finding by Bawden et al. (2018) that multi-encoder architectures, specifically *s-hier-to-2(.tied)*, can outperform a simple concatenation system in the translation of coreferential pronouns.

The Transformer-based models perform strongest on pronouns with intra-segmental antecedent, outperforming the recurrent baseline by 9–18 percentage points. This is likely an effect of increased model depth and the self-attentional architecture in this set of experiments. The model by (Voita et al., 2018) only uses source context, and outperforms the most comparable RNN system, *s-hier.tied*. However, the Transformer-based *concat22* slightly underperforms the RNN-based *concat22*, and we consider it future research how to better exploit target context with Transformer-based models.

## 7 Conclusions

We present a large-scale test suite to specifically test the capacity of NMT models to translate pronouns correctly. The test set contains 12,000 difficult cases of pronoun translations from English *it* to its German counterparts *er, sie* and *es*, extracted automatically from OpenSubtitles (Lison and Tiedemann, 2016).

We evaluate recently proposed context-aware models on our test set. Even though the increase in BLEU score is moderate for all context-aware models, the improvement in the translation of pronouns is considerable: The best model (*s-hier-to-2.tied*) achieves a +16 percentage points gain in accuracy over the baseline.

Our experiments confirm the importance of careful architecture design, with multi-encoder architectures outperforming a model that simply concatenates context sentences. We also demonstrate the effectiveness of parameter sharing between encoders of a context-aware model.

We hope the test set will prove useful for empirically validating novel architectures for context-aware NMT. So far, we have only evaluated models that consider one sentence of context, but the nominal antecedent is more distant for a sizable proportion of the test set, and the evaluation of variable-size context models (Wang et al., 2017; Maruf and Haffari, 2018) is interesting future work.

## Acknowledgements

We are grateful to the Swiss National Science Foundation (SNF) for supporting the project CoNTra (grant number 105212_169888).

## References

Ruchit Agrawal, Turchi Marco, and Negri Matteo. 2018. Contextual Handling in Neural Machine Translation: Look Behind, Ahead and on Both Sides.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating Discourse Phenomena in Neural Machine Translation. In *NAACL 2018*, New Orleans, USA.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia. PMLR.

Liane Guillou. 2016. *Incorporating Pronoun Function into Statistical Machine Translation*. Ph.D. thesis, University of Edinburgh.

Liane Guillou and Christian Hardmeier. 2016. Protest: A test suite for evaluating pronouns in machine translation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Liane Guillou and Christian Hardmeier. 2018. Automatic Reference-Based Evaluation of Pronoun Translation Misses the Point.

Liane Guillou, Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, Mauro Cettolo, Bonnie Webber, and Andrei Popescu-Belis. 2016. Findings of the 2016 wmt shared task on cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation*, pages 525–542, Berlin, Germany. Association for Computational Linguistics.

Christian Hardmeier and Marcello Federico. 2010. Modelling Pronominal Anaphora in Statistical Machine Translation. In *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 283–289.

Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, and Mauro Cettolo. 2015. Pronoun-focused mt and cross-lingual pronoun prediction: Findings of the 2015 discomt shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16, Lisbon, Portugal. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1712.05690*.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does Neural Machine Translation Benefit from Larger Context? *ArXiv e-prints*.

Sébastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Neural machine translation for cross-lingual pronoun prediction. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 54–57, Copenhagen, Denmark. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, San Diego, USA. Ithaca, NY.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Sharid Loáiciga, Sara Stymne, Preslav Nakov, Christian Hardmeier, Jörg Tiedemann, Mauro Cettolo, and Yannick Versley. 2017. Findings of the 2017 discomt shared task on cross-lingual pronoun prediction. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 1–16, Copenhagen, Denmark. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 1275–1284, Melbourne, Australia.

Lesly Miculicich Werlen and Andrei Popescu-Belis. 2017. Validation of an automatic metric for the accuracy of pronoun translation (apt). In *Proceedings of the Third Workshop on Discourse in Machine Translation (DiscoMT)*, EPFL-CONF-229974. Association for Computational Linguistics (ACL).

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Martin Volk, and Gerold Schneider. 2013. Exploiting synergies between open resources for german dependency parsing, pos-tagging, and morphological analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 601–609, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems (NIPS)*, pages 3104–3112.

Jörg Tiedemann and Yves Scherrer. 2017. Neural Machine Translation with Extended Context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Don Tuggener. 2016. *Incremental Coreference Resolution for German*. Ph.D. thesis, University of Zurich.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, page 5998–6008. Curran Associates, Inc.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-Aware Neural Machine Translation Learns Anaphora Resolution. In *ACL 2018*, Melbourne, Australia.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting Cross-Sentence Context for Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark. Association for Computational Linguistics.

# Beyond Weight Tying: Learning Joint Input-Output Embeddings for Neural Machine Translation

Nikolaos Pappas[†] Lesly Miculicich Werlen[†◇] James Henderson[†]
[†]Idiap Research Institute, Martigny, Switzerland
[◇]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{npappas,lmiculicich,jhenderson}@idiap.ch

## Abstract

Tying the weights of the target word embeddings with the target word classifiers of neural machine translation models leads to faster training and often to better translation quality. Given the success of this parameter sharing, we investigate other forms of sharing in between no sharing and hard equality of parameters. In particular, we propose a *structure-aware* output layer which captures the semantic structure of the output space of words within a joint input-output embedding. The model is a generalized form of *weight tying* which shares parameters but allows learning a more flexible relationship with input word embeddings and allows the effective capacity of the output layer to be controlled. In addition, the model shares weights across output classifiers and translation contexts which allows it to better leverage prior knowledge about them. Our evaluation on English-to-Finnish and English-to-German datasets shows the effectiveness of the method against strong encoder-decoder baselines trained with or without *weight tying*.

## 1 Introduction

Neural machine translation (NMT) predicts the target sentence one word at a time, and thus models the task as a sequence classification problem where the classes correspond to words. Typically, words are treated as categorical variables which lack description and semantics. This makes training speed and parametrization dependent on the size of the target vocabulary (Mikolov et al., 2013). Previous studies overcome this problem by truncating the vocabulary to limit its size and mapping out-of-vocabulary words to a single "unknown" token. Other approaches attempt to use a limited number of frequent words plus *sub-word units* (Sennrich et al., 2016), the combination of which can cover the full vocabulary, or to perform character-level modeling (Chung et al., 2016; Lee et al., 2017; Costa-jussà and Fonollosa, 2016; Ling et al., 2015); with the former being the most effective between the two. The idea behind these alternatives is to overcome the vocabulary size issue by modeling the morphology of rare words. One limitation, however, is that semantic information of words or sub-word units learned by the input embedding are not considered when learning to predict output words. Hence, they rely on a large amount of examples per class to learn proper word or sub-word unit output classifiers.

One way to consider information learned by input embeddings, albeit restrictively, is with *weight tying* i.e. sharing the parameters of the input embeddings with those of the output classifiers (Press and Wolf, 2017; Inan et al., 2016) which is effective for language modeling and machine translation (Sennrich et al., 2017; Klein et al., 2017). Despite its usefulness, we find that *weight tying* has three limitations: (a) It biases all the words with similar input embeddings to have a similar chance to be generated, which may not always be the case (see Table 1 for examples). Ideally, it would be better to learn distinct relationships useful for encoding and decoding without forcing any general bias. (b) The relationship between outputs is only implicitly captured by *weight tying* because there is no parameter sharing across output classifiers. (c) It requires that the size of the translation context vector and the input embeddings are the same, which in practice makes it difficult to control the output layer capacity.

In this study, we propose a *structure-aware* output layer which overcomes the limitations of previous output layers of NMT models. To achieve this, we treat words and subwords as units with textual descriptions and semantics. The model consists of a joint input-output embedding which learns what to share between input embeddings

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 73–83
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64008

|  | NMT |  | NMT-tied | NMT-joint |  |
|---|---|---|---|---|---|
| **Query** | **Input** | **Output** | **Input/Output** | **Input** | **Output** |
| visited<br>(Verb past tense) | attacked | visiting | visits | visiting | attended |
|  | conquered | attended | attended | attended | witnessed |
|  | contacted | visit | visiting | visits | discussed |
|  | occupied | visits | frequented | visit | recognized |
|  | consulted | discovered | visit | frequented | demonstrated |
| generous<br>(Adjective) | modest | spacious | generosity | spacious | friendly |
|  | extensive | generosity | spacious | generosity | flexible |
|  | substantial | generously | generously | flexible | brilliant |
|  | ambitious | massive | lavish | generously | fantastic |
|  | sumptuous | huge | massive | massive | massive |
| friend<br>(Noun) | wife | friends | colleague | colleague | colleague |
|  | husband | colleague | friends | friends | fellow |
|  | colleague | Fri@@ | neighbour | neighbour | supporter |
|  | friends | fellow | girlfriend | girlfriend | partner |
|  | painter | friendship | companion | husband | manager |

Table 1: Top-5 most similar input and output representations to two query words based on cosine similarity for an NMT trained without (NMT) or with *weight tying* (NMT-tied) and our *structure-aware* output layer (NMT-joint) on De-En ($|\mathcal{V}| \approx 32K$). Our model learns representations useful for encoding and generation which are more consistent to the dominant semantic and syntactic relations of the query such as verbs in past tense, adjectives and nouns (inconsistent words are marked in red).

and output classifiers, but also shares parameters across output classifiers and translation contexts to better capture the similarity structure of the output space and leverage prior knowledge about this similarity. This flexible sharing allows it to distinguish between features of words which are useful for encoding, generating, or both. Figure 1 shows examples of the proposed model's input and output representations, compared to those of a softmax linear unit with or without *weight tying*.

This proposal is inspired by joint input-output models for zero-shot text classification (Yazdani and Henderson, 2015; Nam et al., 2016a), but innovates in three important directions, namely in learning complex non-linear relationships, controlling the effective capacity of the output layer and handling structured prediction problems.

Our contributions are summarized as follows:

- We identify key theoretical and practical limitations of existing output layer parametrizations such as softmax linear units with or without *weight tying* and relate the latter to joint input-output models.
- We propose a novel *structure-aware* output layer which has flexible parametrization for neural MT and demonstrate that its mathematical form is a generalization of existing output layer parametrizations.
- We provide empirical evidence of the superiority of the proposed structure-aware output layer on morphologically simple and complex languages as targets, including under challenging conditions, namely varying vocabulary sizes, architecture depth, and output frequency.

The evaluation is performed on 4 translation pairs, namely English-German and English-Finnish in both directions using BPE (Sennrich et al., 2016) of varying operations to investigate the effect of the vocabulary size to each model. The main baseline is a strong LSTM encoder-decoder model with 2 layers on each side (4 layers) trained with or without *weight tying* on the target side, but we also experiment with deeper models with up to 4 layers on each side (8 layers). To improve efficiency on large vocabulary sizes we make use of negative sampling as in (Mikolov et al., 2013) and show that the proposed model is the most robust to such approximate training among the alternatives.

## 2 Background: Neural MT

The translation objective is to maximize the conditional probability of emitting a sentence in a target language $Y = \{y_1, ..., y_n\}$ given a sentence in a source language $X = \{x_1, ..., x_m\}$, noted $p_\Theta(Y|X)$, where $\Theta$ are the model parameters learned from a parallel corpus of length $N$:

$$\max_\Theta \frac{1}{N} \sum_{i=1}^{N} \log(p_\Theta(Y^{(i)}|X^{(i)})). \quad (1)$$

By applying the chain rule, the output sequence can be generated one word at a time by calculating the following conditional distribution:

$$p(y_t|y_1^{t-1}, X) \approx f_\Theta(y_1^{t-1}, X). \quad (2)$$

where $f_\Theta$ returns a column vector with an element for each $y_t$. Different models have been proposed to approximate the function $f_\Theta$ (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015; Cho et al., 2014; Gehring et al., 2017; Vaswani et al., 2017). Without loss of generality, we focus here on LSTM-based encoder-decoder model with attention Luong et al. (2015).

## 2.1 Output Layer parametrizations

### 2.1.1 Softmax Linear Unit

The most common output layer (Figure 3a), consists of a linear unit with a weight matrix $W \in \mathrm{I\!R}^{d_h \times |\mathcal{V}|}$ and a bias vector $b \in \mathrm{I\!R}^{|\mathcal{V}|}$ followed by a softmax activation function, where $V$ is the vocabulary, noted as NMT. For brevity, we focus our analysis specifically on the nominator of the normalized exponential which characterizes softmax. Given the decoder's hidden representation $h_t$ with dimension size $d_h$, the output probability distribution at a given time, $y_t$, conditioned on the input sentence $X$ and the previously predicted outputs $y_1^{t-1}$ can be written as follows:

$$\begin{aligned} p(y_t|y_1^{t-1}, X) &\propto \exp(W^T h_t + b) \\ &\propto \exp(W^T I h_t + b), \end{aligned} \quad (3)$$

where $I$ is the identity function. From the second line of the above equation, we observe that there is no explicit output space structure learned by the model because there is no parameter sharing across outputs; the parameters for output class $i$, $W_i^T$, are independent from parameters for any other output class $j$, $W_j^T$.

### 2.1.2 Softmax Linear Unit with *Weight Tying*

The parameters of the output embedding $W$ can be tied with the parameters of the input embedding $E \in \mathrm{I\!R}^{|\mathcal{V}| \times d}$ by setting $W = E^T$, noted as NMT-`tied`. This can happen only when the input dimension of $W$ is restricted to be the same as that of the input embedding ($d = d_h$). This creates practical limitations because the optimal dimensions of the input embedding and translation context may actually be when $d_h \neq d$.

With tied embeddings, the parametrization of the conditional output probability distribution from Eq. 3 can be re-written as:

$$\begin{aligned} p(y_t|y_1^{t-1}, X) &\propto \exp((E^T)^T h_t + b) \\ &\propto \exp(E h_t + b). \end{aligned} \quad (4)$$

As above, this model does not capture any explicit output space structure. However, previous studies have shown that the input embedding learns linear relationships between words similar to distributional methods (Mikolov et al., 2013). The hard equality of parameters imposed by $W = E^T$ forces the model to re-use this implicit structure in the output layer and increases the modeling burden of the decoder itself by requiring it to match this structure through $h_t$. Assuming that the latent linear structure which E learns is of the form $E \approx E_l \mathcal{W}$ where $E_l \in \mathrm{I\!R}^{|\mathcal{V}| \times \mathrm{k}}$ and $\mathcal{W} \in \mathrm{I\!R}^{\mathrm{k} \times \mathrm{d}}$ and $d = d_h$, then Eq. 4 becomes:

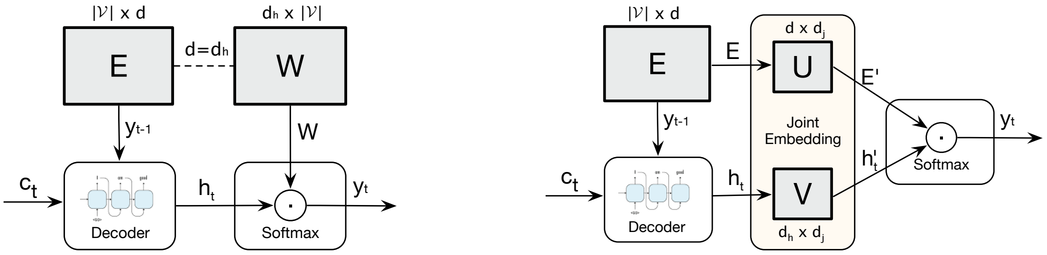
$$p(y_t|y_1^{t-1}, X) \propto exp(E_l \mathcal{W} h_t + b) \,\square. \quad (5)$$

The above form, excluding bias $b$, shows that *weight tying* learns a similar linear structure, albeit implicitly, to joint input-output embedding models with a bilinear form for zero-shot classification (Yazdani and Henderson, 2015; Nam et al., 2016a).[1] This may explain why *weight tying* is more sample efficient than the baseline softmax linear unit, but also motivates the learning of explicit structure through joint input-output models.

## 2.2 Challenges

We identify two key challenges of the existing parametrizations of the output layer: (a) their difficulty in learning complex structure of the output space due to their bilinear form and (b) their rigidness in controlling the output layer capacity due to their strict equality of the dimensionality of the translation context and the input embedding.

[1] The capturing of implicit structure could also apply for the output embedding $W$ in Eq. 3, however that model would not match the bilinear input-output model form because it is based on the input embedding $E$.



(a) Typical output layer which is a softmax linear unit without or with *weight tying* ($W = E^T$).

(b) The *structure-aware* output layer is a joint embedding between translation contexts and word classifiers.

Figure 1: Schematic of existing output layers and the proposed output layer for the decoder of the NMT model with source context vector $c_t$, previous word $y_{t-1} \in \mathrm{I\!R}^{\mathrm{d}}$, and decoder hidden states, $h_t \in \mathrm{I\!R}^{\mathrm{d_h}}$.

### 2.2.1 Learning Complex Structure

The existing joint input-output embedding models (Yazdani and Henderson, 2015; Nam et al., 2016a) have the following bilinear form:

$$E \underbrace{\mathcal{W}}_{Structure} h_t \tag{6}$$

where $\mathcal{W} \in \mathrm{I\!R}^{\mathrm{d \times d_h}}$. We can observe that the above formula can only capture linear relationships between encoded text ($h_t$) and input embedding ($E$) through $\mathcal{W}$. We argue that for structured prediction, the relationships between different outputs are more complex due to complex interactions of the semantic and syntactic relations across outputs but also between outputs and different contexts. A more appropriate form for this purpose would include a non-linear transformation $\sigma(\cdot)$, for instance with either:

$$\text{(a)} \quad \underbrace{\sigma(E\mathcal{W})}_{\text{Output structure}} h_t \quad \text{or} \quad \text{(b)} \; E \underbrace{\sigma(\mathcal{W}h_t)}_{\text{Context structure}} \; . \tag{7}$$

### 2.2.2 Controlling Effective Capacity

Given the above definitions we now turn our focus to a more practical challenge, which is the capacity of the output layer. Let $\Theta_{base}$, $\Theta_{tied}$, $\Theta_{bilinear}$ be the parameters associated with a softmax linear unit without and with *weight tying* and with a joint bilinear input-output embedding, respectively. The capacity of the output layer in terms of effective number of parameters can be expressed as:

$$\mathcal{C}_{base} \approx |\Theta_{base}| = |\mathcal{V}| \times d_h + |\mathcal{V}| \tag{8}$$

$$\mathcal{C}_{tied} \approx |\Theta_{tied}| \leq |\mathcal{V}| \times d_h + |\mathcal{V}| \tag{9}$$

$$\mathcal{C}_{bilinear} \approx |\Theta_{bilinear}| = d \times d_h + |\mathcal{V}|. \tag{10}$$

But since the parameters of $\Theta_{tied}$ are tied to the parameters of the input embedding, the effective number of parameters dedicated to the output layer is only $|\Theta_{tied}| = |\mathcal{V}|$.

The capacities above depend on *external* factors, that is $|\mathcal{V}|$, $d$ and $d_h$, which affect not only the output layer parameters but also those of other parts of the network. In practice, for $\Theta_{base}$ the capacity $d_h$ can be controlled with an additional linear projection on top of $h_t$ (e.g. as in the OpenNMT implementation), but even in this case the parametrization would still be heavily dependent on $|\mathcal{V}|$. Thus, the following inequality for the effective capacity of these models holds true for fixed $|V|$, $d$, $d_h$:

$$\mathcal{C}_{tied} < \mathcal{C}_{bilinear} < \mathcal{C}_{base}. \tag{11}$$

This creates in practice difficulty in choosing the optimal capacity of the output layer which scales to large vocabularies and avoids underparametrization or overparametrization (left and right side of Eq. 11 respectively). Ideally, we would like to be able to choose the effective capacity of the output layer more flexibly moving freely in between $C_{bilinear}$ and $C_{base}$ in Eq. 11.

## 3 Structure-aware Output Layer for Neural Machine Translation

The proposed *structure-aware* output layer for neural machine translation, noted as NMT-`joint`, aims to learn the structure of the output space by learning a joint embedding between translation contexts and output classifiers, as well as, by learning what to share with input embeddings (Figure 1b). In this section, we describe the model in detail, showing how it can be trained efficiently for arbitrarily high number of effective parameters and how it is related to weight tying.

### 3.1 Joint Input-Output Embedding

Let $g_{inp}(h_t)$ and $g_{out}(e_j)$ be two non-linear projections of $d_j$ dimensions of any translation context $h_t$ and any embedded output $e_j$, where $e_j$ is the j$_{th}$ row vector from the input embedding matrix E, which have the following form:

$$e'_j = g_{out}(e_j) = \sigma(Ue_j^T + b_u) \qquad (12)$$

$$h'_t = g_{inp}(h_t) = \sigma(Vh_t + b_v), \qquad (13)$$

where the matrix $U \in \mathbb{R}^{\mathrm{d_j \times d}}$ and bias $b_u \in \mathbb{R}^{\mathrm{d_j}}$ is the linear projection of the translation context and the matrix $V \in \mathbb{R}^{\mathrm{d_j \times d_h}}$ and bias $b_v \in \mathbb{R}^{\mathrm{d_j}}$ is the linear projection of the outputs, and $\sigma$ is a non-linear activation function (here we use `Tanh`). Note that the projections could be high-rank or low-rank for $h'_t$ and $e'_j$ depending on their initial dimensions and the target joint space dimension.

With $E' \in \mathbb{R}^{|\mathcal{V}| \times \mathrm{d_j}}$ being the matrix resulting from projecting all the outputs $e_j$ to the joint space, i.e. $g_{out}(E)$, and a vector $b \in \mathbb{R}^{|\mathcal{V}|}$ which captures the bias for each output, the conditional output probability distribution of Eq 3 can be rewritten as follows:

$$\begin{aligned} & p(y_t|y_1^{t-1}, X) \qquad (14) \\ & \quad \propto \exp\big(E'h'_t + b\big) \\ & \quad \propto \exp\big(g_{out}(E)g_{inp}(h_t) + b\big) \\ & \quad \propto \exp\big(\sigma(UE^T + b_u)\,\sigma(Vh_t + b_v) + b\big). \end{aligned}$$

#### 3.1.1 What Kind of Structure is Captured?

From the above formula we can derive the general form of the joint space which is similar to Eq. 7 with the difference that it incorporates both components for learning output and context structure:

$$\underbrace{\sigma(E\mathcal{W}_o)}_{\text{Output structure}} \; \underbrace{\sigma(\mathcal{W}_c h_t)}_{\text{Context structure}}, \qquad (15)$$

where $\mathcal{W}_o \in \mathbb{R}^{\mathrm{d \times d_j}}$ and $\mathcal{W}_c \in \mathbb{R}^{\mathrm{d_j \times d_h}}$ are the dedicated projections for learning output and context structure respectively (which correspond to $U$ and $V$ projections in Eq. 14). We argue that both nonlinear components are essential and validate this hypothesis empirically in our evaluation by performing an ablation analysis (Section 4.4).

#### 3.1.2 How to Control the Effective Capacity?

The capacity of the model in terms of effective number of parameters ($\Theta_{joint}$) is:

$$\mathcal{C}_{joint} \approx |\Theta_{joint}| = d \times d_j + d_j \times d_h + |\mathcal{V}|. \qquad (16)$$

By increasing the joint space dimension $d_j$ above, we can now move freely between $\mathcal{C}_{bilinear}$ and $\mathcal{C}_{base}$ in Eq .11 without depending anymore on the external factors ($d$, $d_h$, $|V|$) as follows:

$$\mathcal{C}_{tied} < \mathcal{C}_{bilinear} \leq \mathcal{C}_{joint} \leq \mathcal{C}_{base}. \qquad (17)$$

However, for very large number of $d_j$ the computational complexity increases prohibitively because the projection requires a large matrix multiplication between $U$ and $E$ which depends on $|\mathcal{V}|$. In such cases, we resort to sampling-based training, as explained in the next subsection.

### 3.2 Sampling-based Training

To scale up to large output sets we adopt the negative sampling approach from (Mikolov et al., 2013). The goal is to utilize only a sub-set $\mathcal{V}'$ of the vocabulary instead of the whole vocabulary $\mathcal{V}$ for computing the softmax. The sub-set $\mathcal{V}'$ includes all positive classes whereas the negative classes are randomly sampled. During back propagation only the weights corresponding to the subset $\mathcal{V}'$ are updated. This can be trivially extended to mini-batch stochastic optimization methods by including all positive classes from the examples in the batch and sampling negative examples randomly from the rest of the vocabulary.

Given that the joint space models generalize well on seen or unseen outputs (Yazdani and Henderson, 2015; Nam et al., 2016b), we hypothesize that the proposed joint space will be more sample efficient than the baseline NMT with or without *weight tying*, which we empirically validate with a sampling-based experiment in Section 4.5 (Table 2, last three rows with $|\mathcal{V}| \approx 128K$).

### 3.3 Relation to *Weight Tying*

The proposed joint input-output space can be seen as a generalization of *weight tying* ($W = E^T$, Eq. 3), because its degenerate form is equivalent to *weight tying*. In particular, this can be simply derived if we set the non-linear projection functions in the second line of Eq. 14 to be the identity function, $g_{inp}(\cdot) = g_{out}(\cdot) = I$, as follows:

$$\begin{aligned} p(y_t|y_1^{t-1}, X) & \propto \exp\big((IE)\;(Ih_t) + b\big) \\ & \propto \exp\big(Eh_t + b\big)\;\square. \qquad (18) \end{aligned}$$

Overall, this new parametrization of the output layer generalizes over previous ones and addresses their aforementioned challenges in Section 2.2.

|  |  | En → Fi |  | Fi → En |  | En → De |  | De → En |  |
|---|---|---|---|---|---|---|---|---|---|
|  | **Model** | $\|\Theta\|$ | BLEU ($\Delta$) | $\|\Theta\|$ | BLEU ($\Delta$) | $\|\Theta\|$ | BLEU ($\Delta$) | $\|\Theta\|$ | BLEU ($\Delta$) |
| 32K | NMT | 60.0M | 12.68 (–) | 59.8M | 9.42 (–) | 61.3M | 18.46 (–) | 65.0M | 15.85 (–) |
|  | NMT-`tied` | 43.3M | 12.58 (−0.10) | 43.3M | 9.59 (+0.17) | 44.9M | 18.48 (+0.0) | 46.7M | 16.51 (+0.66)† |
|  | NMT-`joint` | 47.5M | **13.03** (+0.35)‡ | 47.5M | **10.19** (+0.77)‡ | 47.0M | **19.79** (+1.3)‡ | 48.8M | **18.11** (+2.26)‡ |
| 64K | NMT | 108.0M | 13.32 (–) | 106.7M | **12.29** (–) | 113.9M | 20.70 (–) | 114.0M | 20.01 (–) |
|  | NMT-`tied` | 75.0M | 13.59 (+0.27) | 75.0M | 11.74 (−0.55)‡ | 79.4M | 20.85 (+0.15) | 79.4M | 19.19 (−0.82)† |
|  | NMT-`joint` | 75.5M | **13.84** (+0.52)‡ | 75.5M | 12.08 (−0.21) | 79.9M | **21.62** (+0.92)‡ | 79.9M | **20.61** (+0.60)† |
| 128K (~) | NMT | 201.1M | 13.52 (–) | 163.1M | 11.64 (–) | 211.3M | 22.48 (–) | 178.3M | 19.12 (–) |
|  | NMT-`tied` | 135.6M | 13.90 (+0.38)* | 103.2M | 11.97 (+0.33)* | 144.2M | 21.43 (−0.0) | 111.6M | 19.43 (+0.30) |
|  | NMT-`joint` | 137.7M | **13.93** (+0.41)† | 103.7M | **12.07** (+0.43)† | 146.3M | **22.73** (+0.25)† | 115.8M | **20.60** (+1.48)‡ |

Table 2: Model performance and number of parameters ($|\Theta|$) with varying BPE operations (32K, 64K, 128K) on the English-Finish and English-German language pairs. The significance of the difference against the NMT baseline with $p$-values $<.05$, $<.01$ and $<.001$ are marked with *, † and ‡ respectively.

## 4 Evaluation

We compare the NMT-`joint` model to two strong NMT baselines trained with and without *weight tying* over four large parallel corpora which include morphologically rich languages as targets (Finnish and German), but also morphologically less rich languages as targets (English) from WMT 2017 (Bojar et al., 2017)[2]. We examine the behavior of the proposed model under challenging conditions, namely varying vocabulary sizes, architecture depth, and output frequency.

### 4.1 Datasets and Metrics

The English-Finnish corpus contains 2.5M sentence pairs for training, 1.3K for development (Newstest2015), and 3K for testing (Newstest2016), and the English-German corpus 5.8M for training, 3K for development (Newstest2014), and 3K for testing (Newstest2015). We preprocess the texts using the BPE algorithm (Sennrich et al., 2016) with 32K, 64K and 128K operations. Following the standard evaluation practices in the field (Bojar et al., 2017), the translation quality is measured using BLEU score (Papineni et al., 2002) (*multi-blue*) on *tokenized* text and the significance is measured with the paired bootstrap re-sampling method proposed by (Koehn et al., 2007).[3] The quality on infrequent words is measured with METEOR (Denkowski and Lavie, 2014) which has originally been proposed to measure performance on function words. To adapt it for our purposes on English-German pairs ($|\mathcal{V}| \approx 32K$), we set as *function words* different sets of words grouped according to three frequency bins, each of them containing $\frac{|\mathcal{V}|}{3}$ words of *high*, *medium* and *low* frequency respectively and set its parameters to $\{0.85, 0.2, 0.6, 0.\}$ and $\{0.95, 1.0, 0.55, 0.\}$ when evaluating on English and German respectively.

### 4.2 Model Configurations

The baseline is an encoder-decoder with 2 stacked LSTM layers on each side from OpenNMT (Klein et al., 2017), but we also experiment with varying depth in the range $\{1, 2, 4, 8\}$ for German-English. The hyperparameters are set according to validation accuracy as follows: maximum sentence length of 50, 512-dimensional word embeddings and LSTM hidden states, dropout with a probability of 0.3 after each layer, and Adam (Kingma and Ba, 2014) optimizer with initial learning rate of 0.001. The size of the joint space is also selected on validation data in the range $\{512, 2048, 4096\}$. For efficiency, all models on corpora with $\mathcal{V} \approx 128K$ ($\sim$) and all *structure-aware* models with $d_j \geq 2048$ on corpora with $\mathcal{V} \leq 64K$ are trained with 25% negative sampling.[4]

### 4.3 Translation Performance

Table 2 displays the results on four translation sets from English-German and English-Finish language pairs when varying the number of BPE operations. The NMT-`tied` model outperforms the

[2] `http://www.statmt.org/wmt17/`

[3] `multi-bleu.perl` and `bootstrap-hypothesis-difference-significance.pl` scripts.

[4] Training the models with a full 128K vocabulary without sampling runs out of memory on our machines.

|  | Model | Layer form | BLEU | $\|\Theta\|$ |
|---|---|---|---|---|
|  | NMT | $W^T h_t$ | 15.85 | 65.0M |
|  | NMT-`tied` | $E h_t$ | 16.51 | 46.7M |
| NMT-`joint` | Eq. 6 | $\mathcal{E}\mathcal{W} h_t$ | 16.23 | 47.0M |
|  | Eq. 7 a | $\sigma(\mathcal{E}\mathcal{W}) h_t$ | 16.01 | 47.0M |
|  | Eq. 7 b | $\mathcal{E}\sigma(\mathcal{W} h_t)$ | 17.52 | 47.0M |
|  | Eq. 15 (512) | $\sigma(\mathcal{E}\mathcal{W}_o)\sigma(\mathcal{W}_c h_t)$ | 17.54 | 47.2M |
|  | Eq. 15 (2048) | $\sigma(\mathcal{E}\mathcal{W}_o)\sigma(\mathcal{W}_c h_t)$ | **18.11** | 48.8M |

Table 3: BLEU scores on De → En ($|\mathcal{V}| \approx 32K$) for the ablation analysis of NMT-`joint`.

NMT baseline in many cases, but the differences are not consistent and it even scores significantly lower than NMT baseline in two cases, namely on Fi → En and De → En with $\mathcal{V} \approx 64K$. This validates our claim that the parametrization of the output space of the original NMT is not fully redundant, otherwise the NMT-`tied` would be able to match its BLEU in all cases. In contrast, the NMT-`joint` model outperforms consistently both baselines with a difference up to +2.2 and +1.6 BLEU points respectively,[5] showing that the NMT-`tied` model has a more effective parametrization and retains the advantages of both baselines, namely sharing weights with the input embeddings, and dedicating enough parameters for generation.

Overall, the highest scores correlate with a high number of BPE operations, namely 128K, 64K, 128K and 64k respectively. This suggests that the larger the vocabulary the better the performance, especially for the morphologically rich target languages, namely En → Fi and En → De. Lastly, the NMT baseline seems to be the least robust to sampling since its BLEU decreases in two cases. The other two models are more robust to sampling, however the difference of NMT-`tied` with the NMT is less significant than that of NMT-`joint`.

### 4.4 Ablation Analysis

To demonstrate whether all the components of the proposed joint input-output model are useful and to which extend they contribute to the performance, we performed an ablation analysis; the results are displayed in Table 3. Overall, all the variants of the NMT-`joint` outperform the baseline with varying degrees of significance. The NMT-`joint` with a bilinear form (Eq. 6) as in (Yazdani and Henderson, 2015; Nam et al., 2016b) is slightly behind the NMT-`tied` and outperforms the NMT baseline; this supports our theoretical analysis in Section 2.1.2 which demonstrated that *weight tying* is learning an implicit linear structure similar to bilinear joint input-output models.

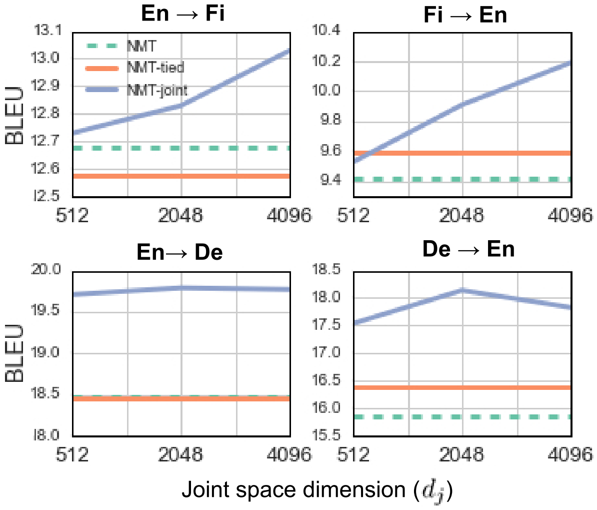


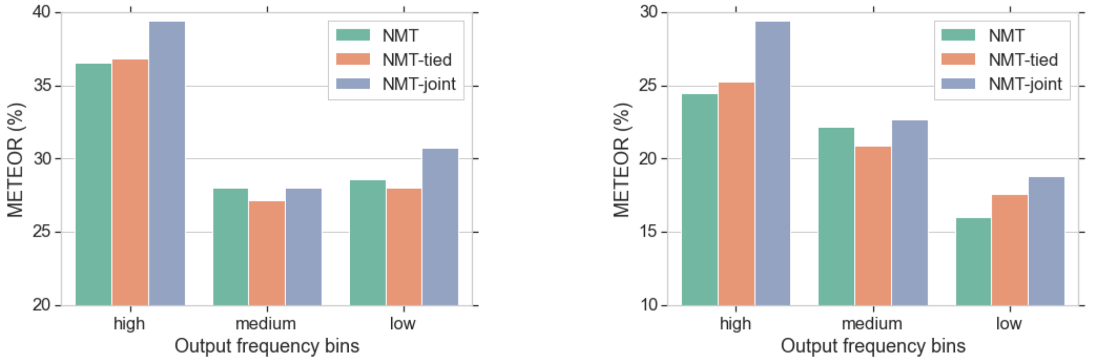
Figure 2: BLEU scores for the NMT-`joint` model when varying its dimension ($d_j$) with $|\mathcal{V}| \approx 32K$.

The NMT-`joint` model without learning explicit translation context structure (Eq. 7 a) performs similar to the bilinear model and the NMT-`tied` model, while the NMT-`joint` model without learning explicit output structure (Eq. 7 b) outperforms all the previous ones. When keeping same capacity (with $d_j$=512), our full model, which learns both output and translation context structure, performs similarly to the latter model and outperforms all the other baselines, including joint input-output models with a bilinear form (Yazdani and Henderson, 2015; Nam et al., 2016b). But when the capacity is allowed to increase (with $d_j$=2048), it outperforms all the other models. Since both nonlinearities are necessary to allow us to control the effective capacity of the joint space, these results show that both types of structure induction are important for reaching the top performance with NMT-`joint`.

### 4.5 Effect of Embedding Size

**Performance** Figure 2 displays the BLEU scores of the proposed model when varying the size of the joint embedding, namely $d_j \in \{512, 2048, 4096\}$, against the two baselines. For English-Finish pairs, the increase in embedding size leads to a consistent increase in BLEU in favor of the NMT-`joint` model. For the English-German pairs, the difference with the baselines is much more evident

[5] Except in the case of Fi → En with $|\mathcal{V}| \approx 64K$, where the NMT baseline performed the best.



(a) Results on En → De ($|\mathcal{V}| \approx 32K$).

(b) Results on De → En ($|\mathcal{V}| \approx 32K$).

Figure 3: METEOR scores (%) on both directions of German-English language pair for all the models when focusing the evaluation on different frequency outputs grouped into three bins (high, medium, low).

| | | Sampling | | |
|---|---|---|---|---|
| **Model** | $d_j$ | 50% | 25% | 5% |
| NMT | - | 4.3K | 5.7K | 7.1K |
| NMT-tied | - | 5.2K | 6.0K | 7.8K |
| NMT-joint | 512 | 4.9K | 5.9K | 7.2K |
| NMT-joint | 2048 | 2.8K | 4.2K | 7.0K |
| NMT-joint | 4096 | 1.7K | 2.9K | 6.0K |

Table 4: Target tokens processed per second during training with negative sampling on En → De pair with a large BPE vocabulary $|\mathcal{V}| \approx 128K$.

and the optimal size is observed around 2048 for De → En and around 512 on En → De. The results validate our hypothesis that there is parameter redundancy in the typical output layer. However the ideal parametrization is data dependent and is achievable systematically only with the joint output layer which is capacity-wise in between the typical output layer and the tied output layer.

**Training speed** Table 4 displays the target tokens processed per second by the models on En → DE with $|\mathcal{V}| \approx 128K$ using different levels of negative sampling, namely 50%, 25%, and 5%. In terms of training speed, the 512-dimensional NMT-joint model is as fast as the baselines, as we can observe in all cases. For higher dimensions of the joint space, namely 2048 and 4096 there is a notable decrease in speed which is remidiated by reducing the percentage of the negative samples.

### 4.6 Effect of Output Frequency and Architecture Depth

Figure 3 displays the performance in terms of METEOR on both directions of German-English language pair when evaluating on outputs of different frequency levels (high, medium, low) for all the competing models. The results on De → EN show that the improvements brought by the NMT-joint model against baselines are present consistently for all frequency levels including the low-frequency ones. Nevertheless, the improvement is most prominent for high-frequency outputs, which is reasonable given that no sentence filtering was performed and hence frequent words have higher impact in the absolute value of METEOR. Similarly, for En → De we can observe that NMT-joint outperforms the others on high-frequency and low-frequency labels while it reaches parity with them on the medium-frequency ones.

We also evaluated our model in another challenging condition in which we examine the effect of the NMT architecture depth in the performance of the proposed model. The results are displayed in Table 5. The results show that the NMT-joint outperforms the other two models consistently when varying the architecture depth of the encoder-decoder architecture. The NMT-joint overall is much more robust than NMT-tied and it outperforms it consistently in all settings. Compared to the NMT which is overparametrized the improvement even though consistent it is smaller for layer depth 3 and 4. This happens because NMT has a much higher number of parameters than NMT-joint with $d_j$=512.

Increasing the number of dimensions $d_j$ of the joint space should lead to further improvements, as shown in Fig. 2. In fact, our NMT-joint with $d_j = 2048$ reaches 18.11 score with a 2-layer deep model, hence it outperforms all other NMT and NMT-tied models even with a deeper architecture (3-layer and 4-layer) regardless of the fact that it utilizes fewer parameters than them (48.8M vs 69.2-73.4M and 50.9-55.1M respectively).

| Model | $d_j$ | 1-layer | $\|\Theta\|$ | 2-layer | $\|\Theta\|$ | 3-layer | $\|\Theta\|$ | 4-layer | $\|\Theta\|$ |
|---|---|---|---|---|---|---|---|---|---|
| NMT | - | 16.49 | 60.8M | 15.85 | 65.0M | 17.71 | 69.2M | 17.74 | 73.4M |
| NMT-tied | - | 15.93 | 42.5M | 16.51 | 46.7M | 17.72 | 50.9M | 17.60 | 55.1M |
| NMT-joint | 512 | **16.93** | 43.0M | **17.54** | 47.2M | **17.83** | 51.4M | **18.13** | 55.6M |

Table 5: BLEU scores on De → En ($|\mathcal{V}| \approx 32K$) for the NMT-joint with $d_j = 512$ against baselines when varying the depth of both the encoder and the decoder of the NMT model.

## 5 Related Work

Several studies focus on learning joint input-output representations grounded to word semantics for zero-shot image classification (Weston et al., 2011; Socher et al., 2013; Zhang et al., 2016), but there are fewer such studies for NLP tasks. (Yazdani and Henderson, 2015) proposed a zero-shot spoken language understanding model based on a bilinear joint space trained with hinge loss, and (Nam et al., 2016b), proposed a similar joint space trained with a WARP loss for zero-shot biomedical semantic indexing. In addition, there exist studies which aim to learn output representations directly from data such as (Srikumar and Manning, 2014; Yeh et al., 2018; Augenstein et al., 2018); their lack of semantic grounding to the input embeddings and the vocabulary-dependent parametrization, however, makes them data hungry and less scalable on large label sets. All these models, exhibit similar theoretical limitations as the softmax linear unit with *weight tying* which were described in Sections 2.2.

To our knowledge, there is no existing study which has considered the use of such joint input-output labels for neural machine translation. Compared to previous joint input-label models our model is more flexible and not restricted to linear mappings, which have limited expressivity, but uses non-linear mappings modeled similar to energy-based learning networks (Belanger and McCallum, 2016). Perhaps, the most similar embedding model to ours is the one by (Pappas and Henderson, 2018), except for the linear scaling unit which is specific to sigmoidal linear units designed for multi-label classification problems and not for structured prediction, as here.

## 6 Conclusion and Perspectives

We proposed a re-parametrization of the output layer for the decoder of NMT models which is more general and robust than a softmax linear unit with or without *weight tying* with the input word embeddings. Our evaluation shows that the *structure-aware* output layer outperforms *weight tying* in all cases and maintains a significant difference with the typical output layer without compromising much the training speed. Furthermore, it can successfully benefit from training corpora with large BPE vocabularies using negative sampling. The ablation analysis demonstrated that both types of structure captured by our model are essential and complementary, as well as, that their combination outperforms all previous output layers including those of bilinear input-output embedding models. Our further investigation revealed the robustness of the model to sampling-based training, translating infrequent outputs and to varying architecture depth.

As future work, the *structure-aware* output layer could be further improved along the following directions. The computational complexity of the model becomes prohibitive for a large joint projection because it requires a large matrix multiplication which depends on $|\mathcal{V}|$; hence, we have to resort to sampling based training relatively quickly when gradually increasing $d_j$ (e.g. for $d_j >= 2048$). A more scalable way of increasing the output layer capacity could address this issue, for instance, by considering multiple consecutive additive transformations with small $d_j$. Another useful direction would be to use more advanced output encoders and additional external knowledge (contextualized or generically defined) for both words and sub-words. Finally, to encourage progress in joint input-output embedding learning for NMT, our code is available on Github: http://github.com/idiap/joint-embedding-nmt.

## Acknowledgments

We are grateful for the support from the European Union through its Horizon 2020 program in the SUMMA project n. 688139, see http://www.summa-project.eu and for the valuable feedback from the anonymous reviewers.

## References

Isabelle Augenstein, Sebastian Ruder, and Anders Sgaard. 2018. Multi-task learning of pairwise sequence classification tasks over disparate label spaces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1896–1906, New Orleans, Louisiana. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, USA.

David Belanger and Andrew McCallum. 2016. Structured prediction energy networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 983–992, New York, New York, USA. PMLR.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. 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)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany. Association for Computational Linguistics.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2016. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL (Demo and Poster Sessions)*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016a. All-in text: learning document, label, and word representations jointly. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016b. All-in text: Learning document, label, and word representations jointly. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 1948–1954, Phoenix, AR, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Nikolaos Pappas and James Henderson. 2018. Joint input-label embedding for neural text classification. *arXiv pre-print arXiv:1806.06219*.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the ACL (Vol. 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 935–943, Lake Tahoe, Nevada.

Vivek Srikumar and Christopher D. Manning. 2014. Learning distributed representations for structured output prediction. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3266–3274, Cambridge, MA, USA. MIT Press.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. WSABIE: Scaling up to large vocabulary image annotation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2764–2770. AAAI Press.

Majid Yazdani and James Henderson. 2015. A model of zero-shot learning of spoken language understanding. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 244–249.

Chih-Kuan Yeh, Wei-Chieh Wu, Wei-Jen Ko, and Yu-Chiang Frank Wang. 2018. Learning deep latent spaces for multi-label classification. In *In Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, USA.

Yang Zhang, Boqing Gong, and Mubarak Shah. 2016. Fast zero-shot image tagging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA.

# A neural interlingua for multilingual machine translation

**Yichao Lu**[*]**, Phillip Keung**[*]**, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, Jason Sun**
{yichaolu,keung,faisall,vikab,shaonanz,jasun}@amazon.com
Amazon Inc.

## Abstract

We incorporate an explicit neural interlingua into a multilingual encoder-decoder neural machine translation (NMT) architecture. We demonstrate that our model learns a language-independent representation by performing direct zero-shot translation (without using pivot translation), and by using the source sentence embeddings to create an English Yelp review classifier that, through the mediation of the neural interlingua, can also classify French and German reviews. Furthermore, we show that, despite using a smaller number of parameters than a pairwise collection of bilingual NMT models, our approach produces comparable BLEU scores for each language pair in WMT15.

## 1 Introduction

### 1.1 Multilingual Machine Translation

Neural machine translation (NMT) relies on word and sentence embeddings to encode the semantic information needed for translation. The standard attentional encoder-decoder models (Bahdanau et al., 2015) for bilingual NMT decompose naturally into separate encoder and decoder subnetworks for the source and target languages. This factorization has inspired various forms of multilingual NMT models that extended the original bilingual framework to handle more language pairs simultaneously. We refer to NMT models that accept sentences from one source language and produce outputs in one target language as ‘bilingual’. We contrast this with ‘multilingual’ NMT models, which support more than one source and/or target languages within the same model.

The naive approach to multilingual machine translation would train a model for each language pair, which scales quadratically with the number of languages in the corpus. Instead, by combining language-specific encoders and decoders in different ways, Dong et al. (2015), Zoph and Knight (2016), Luong et al. (2016), and Firat et al. (2016a) have explored the one source-to-many target, many source-to-one target, and many source-to-many target multilingual MT settings. The multi-way shared attention model (Firat et al., 2016a) is closest to our work, in that they consider the large-scale, many-to-many scenario with multiple encoders and decoders.

It is also possible to adapt existing bilingual NMT models to the many-to-many case without changing the architecture at all. The universal encoder-decoder approach (Ha et al., 2016; Johnson et al., 2017) constructs a shared vocabulary for all languages in the dataset, and use just one encoder and decoder for multilingual translation. In addition, Johnson et al. (2017) introduce *direct zero-shot translation*, which refers to the task of translating between language pairs without parallel text or pivoting through an intermediate language like English. Direct zero-shot translation may yield lower BLEU scores than pivot-based approaches, but avoids doubling the latency and computational overhead (due to translating the source sentence twice,) which is a concern for large-scale, productionized MT systems.

Nonetheless, both the multi-way shared attention model and the universal encoder-decoder model suffer from certain disadvantages. For the former, direct zero-shot translation was shown to be impossible in Firat et al. (2016b), and there is no indication that the model learns any kind of shared representation across languages. For the latter, the output vocabulary size is typically fixed to the vocabulary size for a single target language (i.e. roughly 20,000 to 30,000 types), regardless of the number of languages in the corpus. Increasing the vocabulary size is costly, since the training

[*] Equal contribution

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 84–92
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64009

and inference time scales linearly with the size of the decoder's output layer.

### 1.2 Our Contributions

In this work, we construct an explicit *neural interlingua* for multilingual NMT, which addresses some of the limitations in existing approaches. Our contributions are threefold:

Firstly, we describe an attentional neural interlingua that receives language-specific encoder embeddings and produces output embeddings which are agnostic to the source and target languages.

Secondly, we perform zero-shot translation (without pivot translation) for the Fr↔Ru, Zh↔Es and Es↔Fr pairs of the updated UN Parallel Corpus (Ziemski et al., 2016). At the time of writing, our approach is the only alternative to the universal encoder-decoder model for direct neural zero-shot translation. We observe a significant improvement in zero-shot translation performance compared to that model.

Finally, we demonstrate that our model generates useful representations for crosslingual transfer learning. We use the source sentence embeddings from our translation model to create an English Yelp review classifier that can, through the mediation of the interlingua, classify French and German Yelp reviews. We also show that the sentence embeddings of parallel translations are close to each other in a low-dimensional space.

## 2 Model Architecture

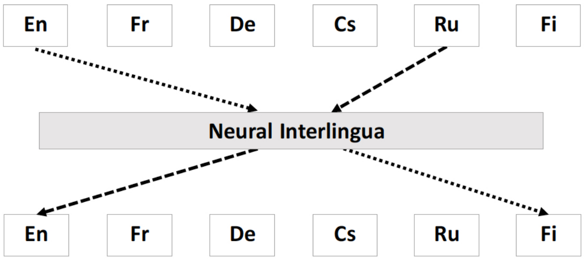


Figure 1: Our encoder-decoder model with the neural interlingua, trained on WMT15. The neural interlingua is an attentional encoder that converts language-specific embeddings to language-independent ones. Here, we illustrate the flow of data from English → Interlingua → Finnish, and Russian → Interlingua → English.

Figure 1 illustrates our basic model architecture. Each language has its own recurrent encoder and decoder. We attempt to construct a neural interlingua by passing the language-specific encoder embeddings through a shared recurrent layer, whose output embeddings are then passed to language-specific decoders.

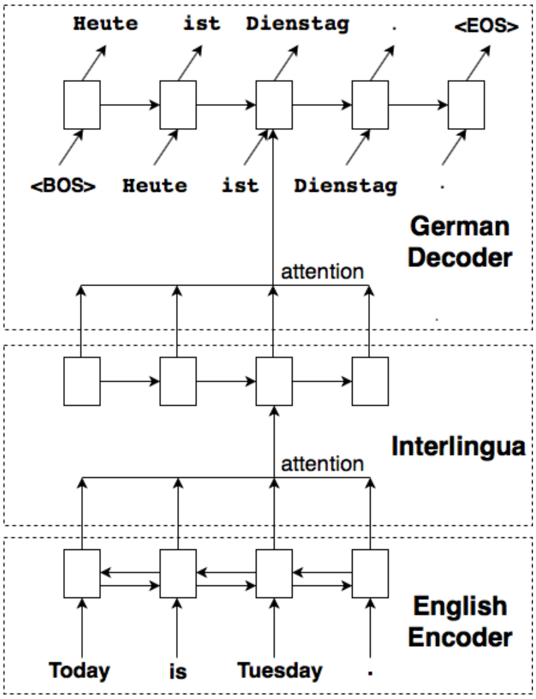


Figure 2: An in-depth look at the network structure when training/predicting with an En-De batch. The English sentence is fed through the English bidirectional LSTM encoder. The encoder states are passed into the neural interlingua, which is an attentional LSTM encoder. Finally, the hidden states of the interlingua are consumed by the German attentional LSTM decoder to generate the German translation.

The figure describes the flow of data in the model; each minibatch only contains one source language and one target language, and only the parameters in the source encoder, interlingua, and target decoder are used for the forward and backward passes. During training, the source and target languages in each minibatch rotate according to a schedule (see Algorithm 1). In Figure 2, we illustrate how an English sentence is converted into a German one.

As with most sequence-to-sequence models, we can view the generation of the next token in the target sentence as the application of a series of neural network operations on the source sentence and the partial output thus far. We model the probability of each target sentence as follows,

$$\begin{aligned} p(y_i|y_{<i}, x) = \text{Dec}_t(&\text{Inter}(\text{Enc}_s(\text{Emb}_s(x))), \\ &y_{i-1}, h^t_{i-1}) \end{aligned}$$

where $y$ is the target sentence, $x$ is the source sentence, Dec$_t$ is the decoder for the target language $t$, Inter is the neural interlingua, Enc$_s$ is the encoder for the source language $s$, Emb$_s$ is the word embedding matrix for $s$, $h^t_{i-1}$ is the state of the decoder at step $i-1$, $s \in \{1, ..., S\}$ is the index of the source language, and $t \in \{1, ..., T\}$ is the index of the target language.

The source sentence $x$ is transformed from a sequence of one-hot representations to a sequence of word embeddings $B^s$ through Emb$_s$,

$$B^s = \text{Emb}_s(x^s)$$

$B^s$ is a $b^s \times L_x$ matrix, where $L_x$ is the length of the source sentence, and $b^s$ is the size of the word embedding for the source language $s$.

The sequence of word embeddings is converted into a sentence representation $E^s$ by Enc$_s$,

$$\begin{aligned} E^s_{.,i} &= \text{Enc}_s(B^s)_{.,i} \\ &= \text{BiLSTM}(B^s_{.,i}, h^s_{i-1}) \end{aligned}$$

$E^s$ is a $e^s \times L_x$ matrix, where $e^s$ is the size of encoder's output. The notation $X_{.,i}$ refers to the $i^{\text{th}}$ column of the matrix $X$. BiLSTM is a bidirectional LSTM network, with forward and backward states $h^s_{i-1} = [\overrightarrow{h}^s_{i-1}, \overleftarrow{h}^s_{i+1}]$ for step $i-1$.

The neural interlingua Inter is an attentional encoder that maps the language-specific representation $E^s$ to an interlingual representation $I$,

$$\begin{aligned} I_{.,i} &= \text{Inter}(E^s)_{.,i} \\ &= W^I[\text{LSTM}(c^I_i, h^I_{i-1}), c^I_i] + b^I \\ &= W^I[h^I_i, c^I_i] + b^I \end{aligned}$$

where $h^I_{i-1}$ is the interlingua LSTM state for step $i-1$, $c^I_i = \sum_{j=1}^{L_x} \alpha^I_{ij} E^s_{.,j}$ is the attentional context vector, $\alpha^I_{ij} = \frac{exp(e^I_{ij})}{\sum_j exp(e^I_{ij})}$ and $e^I_{ij} =$ MLP$_I(h^I_i, E^s_{.,j})$ are the normalized and unnormalized attention weights introduced in Bahdanau et al. (2015), and $z = [x, y]$ denotes the concatenation of the vectors $x$ and $y$ into a new vector $z$. We perform an affine transformation with $W^I, b^I$ to project the interlingua output to the desired dimensions.

$I$ is a $e^i \times L_i$ matrix, where $e^i$ is the size of the interlingua's output. The output of the neural interlingua is always fixed in length to $L_i$ (where $L_i = 50$ in our experiments), regardless of the length of the source sentence. We chose $L_i = 50$ because, during model training, we restrict the maximum source sentence length to 50. To avoid learning language-specific embeddings, we do not use indicator tokens for the source or target languages.

Finally, the decoder takes the interlingual representation $I$ and the partial target sentence $y_{<i}$ and computes the probability distribution for the next output token,

$$\begin{aligned} & p(y_i|y_{<i}, x) \\ & = \text{Dec}_t(I, y_{i-1}, h^t_{i-1})_{.,i} \\ & = \text{softmax}(W^t[\text{LSTM}([y_{i-1}, c^t_i], h^t_{i-1}), c^t_i] + b^t) \\ & = \text{softmax}(W^t[h^t_i, c^t_i] + b^t) \end{aligned}$$

where $c^t_i = \sum_{j=1}^{L_i} \alpha^t_{ij} I_{.,j}$ is the context vector at step $i$, and $\alpha^t_{ij}$ are the normalized attention weights. The decoders receive the source sentence only through the interlingual embedding.

Like Firat et al. (2016a), the number of encoders and decoders for our model architecture scales linearly (rather than quadratically) with the number of languages. In addition, since the neural interlingua provides a common source sentence representation to all decoders, the number of attention mechanisms also scales linearly with the number of languages.

We note that the concept of a neural interlingua is independent of the architecture that is chosen. While we use a LSTM encoder-decoder model with single-headed attention for experimental simplicity, one could also introduce a neural interlingua to a transformer network (Vaswani et al., 2017) or a CNN encoder-decoder network (Gehring et al., 2017) instead.

## 3 Experiments

We conducted 4 experiments with our model.

We compared the performance of bilingual NMT baselines against our proposed multilingual model, and observe comparable performance across all the language pairs in WMT15.

| Parameter | Multi<br>-lingual | Bilingual |
|---|---|---|
| vocabulary size | 30,000 | 30,000 |
| source embedding size | 256 | 256 |
| target embedding size | 256 | 256 |
| output dimension | 512 | 512 |
| encoder hidden size | 512 | 512 |
| decoder hidden size | 512 | 512 |
| interlingua hidden size | 512 | - |
| interlingua length | 50 | - |
| encoder depth | 2 | 4 |
| interlingua depth | 1 | 0 |
| decoder depth | 1 | 1 |
| attention type | additive | additive |
| optimizer | Adam | Adam |
| learning rate | 0.0002 | 0.0002 |
| batch size | 400 | 400 |

Table 1: Hyperparameters for the multilingual and bilingual encoder-decoder models.

We found that the language-independent sentence embeddings can be used for zero-shot multilingual classification. We train an English Yelp review classifier with the interlingual embeddings as input features, and use that model to classify French and German reviews.

We performed direct zero-shot translation for 3 language pairs in the new UN Parallel Corpus. For this task, our model showed an improvement over the model architecture described in Johnson et al. (2017). Our positive experimental finding confirms that our model provides a new approach for direct neural zero-shot translation.

Finally, we visualized the language-independent sentence embeddings by projecting them down to 2 dimensions. We observe that parallel translations of French, German and English sentences remain close to each other in this low-dimensional space.

### 3.1 Model Training

The hyperparameters for the bilingual baseline models and our multilingual network are summarized in Table 1. Our multilingual model uses 1 bidirectional LSTM layer in the encoder for each input language, 1 attentional LSTM layer for the interlingua and 1 attentional LSTM layer in the decoder for each output language. The baseline bilingual models use 2 bidirectional LSTM layers in the encoder and 1 attentional LSTM layer in the decoder. We chose the Adam optimizer (Kingma and Ba, 2015), and we used importance sampling, as described in Jean et al. (2015), to accelerate model training.

### 3.2 Language Rotation During Training

**Algorithm 1:** Multilingual model training schedule on WMT15. We store the cycle of language pairs in $schedule$, and $x_s$ and $y_t$ refer to the source and target sentences respectively.

```
$\theta \leftarrow$ RandomInitializer()
$schedule \leftarrow \{\}$
for $S \in \{En, Fr, De, Cs, Fi, Ru\}$ do
    for $L \in \{Fr, De, Cs, Fi, Ru\}$ do
        $schedule \mathrel{+}= \{(En, L), (L, En)\}$
    end
    $schedule \mathrel{+}= \{(S, S)\}$
end
while $True$ do
    for $(s, t) \in schedule$ do
        $x_s \leftarrow SampleSource(s)$
        $y_t \leftarrow SampleTarget(t)$
        $a \leftarrow ForwardStep(\theta, x_s, y_t)$
        $\nabla\theta \leftarrow BackwardStep(a, \theta)$
        $\theta \leftarrow SGDUpdate(\theta, \nabla\theta)$
    end
end
```

The language pair schedule used during training is crucial for learning an effective sentence representation. We provide the details in Algorithm 1. In our initial experiments, we cycled through 10 language pairs (i.e. ($x$ → En, En → $x$), $x \in$ {Fr, De, Ru, Cs, Fi}), where each minibatch consisted of sentences from one language pair. However, we found that the naive schedule failed to produce a useful representation for zero-shot translation or crosslingual text classification. Since WMT15 is not a multi-parallel corpus, the model essentially learns to handle two separate tasks, namely translation from English and translation to English. For instance, since the output of the De encoder and the En encoder would never be used by the same decoder, there is no reason for De and En source sentences to share the same embedding, even if they are translations of each other.

To encourage the model to share the encoder representations across English and non-English

| Source | Target | Bilingual | Multilingual |
|---|---|---|---|
| En | Fr | **34.85** | 33.80 |
|  | De | **23.67** | 23.37 |
|  | Cs | **17.60** | 16.62 |
|  | Ru | 21.26 | **21.92** |
|  | Fi | 11.55 | **13.34** |
| Fr | En | **30.72** | 30.24 |
| De |  | 27.08 | **27.29** |
| Cs |  | 23.00 | **23.87** |
| Ru |  | 24.14 | **26.15** |
| Fi |  | 14.77 | **16.58** |

Table 2: Comparison of BLEU scores across language pairs in newstest2015 and newsdiscuss2015. We show the results for the bilingual baseline NMT models and our multilingual NMT model.

source sentences, we added an extra identity language pair (i.e. De → De, En → En, etc.) to the rotation. The identity pair forces the source embeddings to be compatible with an additional decoder. We found that when we did not include the identity mapping task during training, the zero-shot BLEU score was $< 1.0$ for the Fr-Ru language pair.

### 3.3 Multilingual NMT versus Bilingual NMT

We used the training corpora from the WMT15 translation task to train our encoder-decoder models. The dataset provides English ↔ (German, French, Czech, Russian, Finnish) parallel sentences. We followed the standard WMT preprocessing recipes[1], which are based on the Moses library (Koehn et al., 2007). For each language, we created a vocabulary of 30,000 word pieces using byte pair encoding (Sennrich et al., 2016). Sentences longer than 50 word pieces were removed from the training corpus. We used newstest2014 and newsdev2015 as our development set, and newstest2015 and newsdiscuss2015 as our test set.

We compared the performance of the multilingual model against bilingual baseline models. The BLEU scores are provided in Table 2. Results are reported on newstest2015 and newsdiscuss2015. We see that, while the performance is broadly similar (i.e. generally <1.0 BLEU) between the our model and the baselines, there is a decrease in BLEU for higher-resource languages (e.g. Fr) and an increase in BLEU for lower-resource languages (e.g. Fi, Ru). We suspect that this is a consequence of the language pair schedule, which cycles through all language pairs as though they were equally frequent in the corpus. A similar effect was also observed in Johnson et al. (2017).

Currey et al. (2017) have shown that (specifically in low-resource settings) using copied monolingual data can improve model performance. We followed the technique in Currey et al. (2017) to strengthen the baseline models, but did not observe an improvement in the final BLEU score. This may be due to the fact that even the smallest language pair in WMT15 has 2 million sentence pairs, which is more than 3 times larger than either the Tr-En or Ro-En pairs discussed in Currey et al. (2017).

As with Firat et al. (2016a), we generally see an improvement when translating to English. We believe that this is because the English language model is stronger in the multilingual case, since the English decoder sees more English text.

### 3.4 Zero-shot Multilingual Classification

We constructed a multilingual Yelp review dataset from a subset of the Yelp Challenge (Round 10) corpus. We restrict ourselves to English, French, and German reviews. The training corpus consists of 5,000 English Yelp reviews, and the test sets contain 4,000 reviews for each language. The French and German reviews were extracted by applying language detection on reviews from Quebec, Canada and Baden-Württemberg, Germany. The review scores were binarized, where 4 and 5 star reviews were labeled as positive, and 1 and 2 star reviews were labeled as negative. We reuse the encoders trained in Section 3.3 in this section's experiments.

At training time, an English Yelp review is treated as one sentence; we do not apply sentence segmentation to the review. It is passed through the English encoder, and the neural interlingua converts the English sentence representation to a fixed-length representation. To create a feature vector for the text classifier, we apply mean-pooling to the sentence representation. Under our experimental settings, every sentence is converted to a $512 \times 50$ interlingual embedding, which is mean-pooled into a 512-dimensional vector. We then fit a logistic regression model using this feature vector and the sentence polarity as the binary label. The classifier is only trained on En-

[1] e.g. http://data.statmt.org/wmt17/translation-task/preprocessed/de-en/prepare.sh

| Color | Lang. | Text |
|---|---|---|
| Green | En | spreads between sovereign bonds in Germany and those in other countries were relatively unaffected by political and market uncertainties concerning Greece in late 2014 and early 2015 . |
| | Fr | par contre , la différence entre les obligations souveraines allemandes et celles d'autres pays a été relativement peu touchée par les incertitudes politiques et les doutes des marchés concernant la Grèce fin 2014 et début 2015 . |
| | Ru | политическая и рыночная нестабильность , связанная с ситуацией в Греции в конце 2014 - го и начале 2015 года , практически не отразилась на спредах доходности между государственными облигациями Германии и других стран . |
| Red | En | 13 . we underscore the need to accelerate efforts at all levels to achieve the objectives of the international arrangement on forests beyond 2015 and the need to establish a stronger , more effective and solid arrangement for the period 2015 to 2030 ; |
| | Fr | 13 . nous soulignons qu'il faudra redoubler d'efforts à tous les niveaux pour atteindre les objectifs de l'arrangement international après 2015 et qu'il faudra mettre en place un arrangement plus solide et plus efficace pour la période 2015 - 2030 ; |
| | Ru | 13 . мы подчеркиваем , что необходимо активизировать усилия на всех уровнях в интересах достижения целей международного механизма по лесам на период после 2015 года и создать действенный , более эффективный и надежный механизм на период 2015 - 2030 годов ; |
| Orange | En | the various training activities are listed in table 2 below . |
| | Fr | on énumère dans le tableau 2 ci - dessous les diverses activités de formation . |
| | Ru | в представленной далее таблице 2 приведен перечень различных мероприятий по профессиональной подготовке . |
| Blue | En | the Conference affirms that , pending the realization of this objective , it is in the interest of the very survival of humanity that nuclear weapons never be used again . |
| | Fr | elle affirme que , en attendant la réalisation de cet objectif , il est dans l'intérêt de la survie même de l'humanité que les armes nucléaires ne soient plus jamais utilisées . |
| | Ru | конференция заявляет , что , пока эта цель не достигнута , необходимо в интересах самого выживания человечества добиться того , чтобы ядерное оружие никогда не было вновь применено . |

Table 3: Text of the parallel sentences in Figure 3.

| | **Input Language** | | |
|---|---|---|---|
| | En | De | Fr |
| **Trigram** | 91.6% ± 0.9% | 89.6% ± 0.9% | 91.5% ± 0.9% |
| **Embeddings** | 91.5% ± 0.9% | 89.2% ± 0.9% | 91.1% ± 0.9% |
| **% Positive** | 82.9% | 86.7% | 88.5% |

Table 4: Accuracy for crosslingual Yelp binary review classification. The trigram baseline model was trained on English reviews, and tested on English reviews and English translations of French and German reviews. The embedding-based classifier uses interlingual embeddings from our model in Section 3.3. '% Positive' refers to the proportion of the test set that has a positive label.

glish reviews.

At prediction time, we pass the text of a German review through the German encoder and the interlingua, which is again mean-pooled to form a 512-dimensional vector. Since the interlingual representation should be language-independent, we can attempt to classify German reviews by providing the vector representation of the German review to the English classifier. We use the same process for French reviews.

In Table 4, we compare the accuracy of the classifier trained on English review embeddings to that of a baseline model. We established the baseline by training a trigram classifier on the English reviews, and used English translations of the French and German reviews for classification. We obtained the translations through the Google Translate API. The classification accuracy using the interlingual embeddings or the translated French and German reviews are similar, which shows that the embeddings have retained semantic information in a language-independent way.

### 3.5 Direct Zero-shot Translation

The updated UN Parallel Corpus (Ziemski et al., 2016), unlike the WMT corpus, is a fully multi-parallel corpus that contains English, Spanish, French, Arabic, Chinese and Russian text. We used this corpus as a testbed for our zero-shot translation experiments.

We trained our multilingual model on the UN corpus, following the same settings that we used

| | Fr-Ru | Ru-Fr | Es-Zh | Zh-Es | Es-Fr | Fr-Es |
|---|---|---|---|---|---|---|
| **This Work** | 18.24 | 21.61 | 17.66 | 18.66 | 30.08 | 31.94 |
| **Univ. Enc-Dec** | 8.77 | 9.76 | 8.62 | 6.13 | 15.04 | 14.37 |
| **Pivot** | 20.87 | 27.34 | 26.03 | 26.01 | 31.84 | 32.93 |
| **Direct NMT** | 28.29 | 33.26 | 32.36 | 32.69 | 41.38 | 44.49 |

Table 5: Zero-shot BLEU scores on the UN Parallel Corpus on selected language pairs. The universal encoder-decoder, pivot and direct NMT results were retrieved from Miura et al. (2017). Our proposed model outperforms the universal encoder-decoder model (Johnson et al., 2017) on the zero-shot translation task.

for the WMT corpus (see Table 1 and Algorithm 1). The text was processed following the steps provided in Miura et al. (2017). We restrict the training corpus to sentence pairs that have English as either the source or target language.

We used the Fr-Ru, Es-Zh and Es-Fr portions of the test set from the UN corpus for the zero-shot translation evaluation. The training dataset that we constructed does not contain direct Fr-Ru, Es-Zh or Es-Fr sentence pairs. The test set contains 4,000 sentence pairs for each language pair.

We examine the BLEU scores for zero-shot translation on the UN corpus in Table 5. The universal encoder-decoder, pivot and direct NMT results were retrieved from (Miura et al., 2017). By 'direct NMT', we refer to a model trained directly on the parallel text.

Our multilingual model performs significantly better on the direct zero-shot task than the universal encoder-decoder approach of Johnson et al. (2017). Generally, our model does not perform as well as the pivot approach, though in the case of Es-Fr and Fr-Es, the difference is surprisingly small (<2.0 BLEU).

Improving direct zero-shot methods to reach parity with pivot translation has practical consequences for large-scale NMT systems, like reduced latency and computational overhead. (Recall that pivot translation must translate every source sentence twice; first into the intermediate language, and then into the target language.) Our results show progress towards the goal of transitioning away from pivot-based methods to neural zero-shot translation.

### 3.6 Interlingua Visualization

In Figure 3, we plot the embeddings for 4 groups of parallel sentences. Sentences from the same group share the same color. Each group contains one French, one English and one Russian sentence which are parallel translations of each other. We provide the text of the embedded sentences in Table 3.

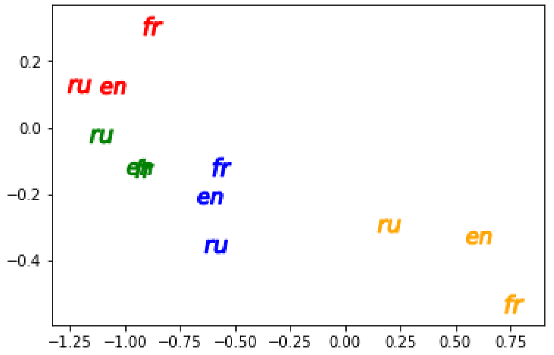


Figure 3: Interlingual embeddings for four groups of parallel English, French, and Russian sentences from the UN Parallel Corpus. The 512-dimensional mean-pooled interlingual sentence embeddings were projected down to $\mathbf{R}^2$ using PCA. Refer to Table 3 for the colors and text of the sentences.

The embeddings were generated by mean-pooling each sentence embedding to a 512-dimensional vector and projecting it to $\mathbf{R}^2$ using PCA. From the figure, we observe a clear separation between different groups of sentences, while sentences within the same group remain close to each other in space. This is the expected outcome if our model has captured language-independent semantic information in its sentence representations.

## 4 Related Work

### 4.1 Networks with Language-specific Encoders and Decoders

The many-to-one approach explored in Zoph and Knight (2016) primarily considers the trilingual case, where a multi-parallel corpus is available, and uses 2 encoders simultaneously to provide the source context for the decoder. We note that using 2 encoders simultaneously requires having 2 source sentences for every desired target sentence at prediction time, which is not the setting that we investigate here.

By combining a single encoder with multiple attentional decoders, the one-to-many approach presented in Dong et al. (2015) showed an improvement in translation performance, due to the increase in the number of sentences seen by the encoder and through multi-task learning.

The many-to-many approach in the shared attention model (Firat et al., 2016a) assigns a different encoder and decoder to each language, but shares the decoders' attention mechanisms. By specifying a 'universal' attention mechanism for all language pairs, Firat et al. (2016a) avoid creating as many attention mechanisms as there are language pairs (i.e. avoids quadratic scaling).

However, the attention mechanism acts as the alignment model between the source and target sentences, and a shared attention mechanism may be too restrictive, especially for languages that have very different word orders. Our interlingual approach relaxes the requirement of a single, shared attention mechanism. In our framework, there are as many attention mechanisms as there are decoders.

### 4.2 Universal Encoder-Decoder Networks

Johnson et al. (2017) have foregone the use of multiple encoders and decoders, and instead use one universal encoder and one universal decoder. They constructed a joint vocabulary for all languages in the corpus, consisting of word pieces derived from a byte-pair encoding (Sennrich et al., 2016) on the union of the vocabulary of all the languages, and include special tokens to indicate what the output language should be. Ha et al. (2016) follow a similar approach, but the shared vocabulary is constructed by prepending a language identifier to each token.

The universal encoder-decoder approach does have some shortcomings. Johnson et al. (2017) rely on the existence of a shared vocabulary, which may not be as sensible in some combinations (e.g. Chinese and English) as in others (e.g. Spanish and Portuguese). If the languages' vocabularies do not share many word pieces, then either the decoder's output layer will be very large, which slows down training and inference, or the output layer will be artificially constrained to a manageable size, which impacts translation performance.

Our approach, on the other hand, allows each target language to retain its own decoder. The total vocabulary size can then expand with the number of languages without affecting training or inference speed.

### 4.3 Zero-shot Translation

One of the challenges in multilingual MT is data sparsity, which refers to the lack of parallel text for every possible language pair in a corpus. Zero-shot translation is the task of translating between language pairs without parallel text.

An early approach to allow zero-shot translation made use of a 'pivot' language in the translation process (Boitet, 1988). For instance, in sentence-based pivoting, the source sentence is translated into a pivot language, and from the pivot language translated to the target language. Various extensions of the pivot technique have been proposed over the years, see Utiyama and Isahara (2007), Chen et al. (2017), Miura et al. (2017), Cohn and Lapata (2007).

Universal encoder-decoder systems like Johnson et al. (2017) have demonstrated the ability to perform direct zero-shot translation without using a pivot language at all, albeit with a significant BLEU reduction for some language pairs.

## 5 Conclusion

We incorporate a neural interlingua component into the standard encoder-decoder framework for multilingual neural machine translation, and demonstrate that the resulting model learns language-independent sentence representations, enabling zero-shot translation and crosslingual text classification.

We perform direct zero-shot translation for 3 language pairs without pivoting through an intermediate language like English. We observe an improvement in zero-shot translation performance compared to the universal encoder-decoder results reported in Miura et al. (2017). Furthermore, we use the learned encoder to train an English Yelp review classifier that can, with the help of the interlingual embeddings, also classify German and French reviews. Finally, our experiments showed that the results from our model are comparable to the results from bilingual baselines.

In future work, we intend to address the significant performance gap between direct neural zero-shot translation and pivot translation. By manipulating the sentence embeddings in an appropriate way, we aim to extract significant improvements over the results presented in this paper.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Christian Boitet. 1988. Pros and cons of the pivot and transfer approaches in multilingual machine translation. In *Readings in machine translation*, pages 273–279.

Yun Chen, Yang Liu, Yong Cheng, and Victor OK Li. 2017. A teacher-student framework for zero-resource neural machine translation. *ACL*.

Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *ACL*.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *WMT*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *ACL*.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL-HLT*.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *EMNLP*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *IWSLT*.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *ACL*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: enabling zero-shot translation. In *TACL*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *ICLR*.

Akiva Miura, Graham Neubig, Katsuhito Sudoh, and Satoshi Nakamura. 2017. Tree as a pivot: Syntactic matching methods in pivot translation. In *WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *NAACL-HLT*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *LREC*.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *NAACL-HLT*.

# Improving Neural Language Models with Weight Norm Initialization and Regularization

**Christian Herold**[*] **Yingbo Gao**[*] **Hermann Ney**
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
<surname>@i6.informatik.rwth-aachen.de

## Abstract

Embedding and projection matrices are commonly used in neural language models (NLM) as well as in other sequence processing networks that operate on large vocabularies. We examine such matrices in fine-tuned language models and observe that a NLM learns word vectors whose norms are related to the word frequencies. We show that by initializing the weight norms with scaled log word counts, together with other techniques, lower perplexities can be obtained in early epochs of training. We also introduce a weight norm regularization loss term, whose hyperparameters are tuned via a grid search. With this method, we are able to significantly improve perplexities on two word-level language modeling tasks (without dynamic evaluation): from 54.44 to 53.16 on Penn Treebank (PTB) and from 61.45 to 60.13 on WikiText-2 (WT2).

## 1 Introduction

A language model (LM) measures how likely a certain sequence of words is for a given language. It does so by calculating the probability of occurrence of that sequence, which can be learned from monolingual text data. Many models in machine translation and automatic speech recognition benefit from the use of a LM (Corazza et al., 1995; Peter et al., 2017).

While count-based LMs (Katz, 1987; Kneser and Ney, 1995) provided the best results in the past, substantial improvements were achieved with the introduction of neural networks in the field of language modeling (Bengio et al., 2003). Different types of architectures such as feedforward neural networks (Schwenk, 2007) and recurrent neural networks (Mikolov et al., 2010) have since been used for language modeling. Currently, variants of long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) networks give the best results on popular language modeling tasks (Yang et al., 2018).

In natural language processing, words are typically represented by high-dimensional one-hot vectors. To reduce dimensionality and to be able to learn relationships between words, they are mapped into a lower-dimensional, continuous embedding space. Mathematically, this is done by multiplying the one-hot vector with the embedding matrix. Similarly, to receive a probability distribution over the vocabulary, a mapping from an embedding space is performed by a projection matrix followed by a softmax operation. These two matrices can be tied together in order to reduce the number of parameters and improve the results of NLMs (Inan et al., 2017; Press and Wolf, 2017).

Since the row vectors in the embedding and projection matrices are effectively word vectors in a continuous space, we investigate such weight vectors in well-trained and fine-tuned NLMs. We observe that the learned word vector generally has a greater norm for a frequent word than an infrequent word. We then specifically examine the weight vector norm distribution and design initialization and normalization strategies to improve NLMs.

Our contribution is twofold:

- We identify that word vectors learned by NLMs have a weight norm distribution that resembles logarithm of the word counts. We then correspondingly develop a weight initialization strategy to aid NLM training.
- We design a weight norm regularization loss term that increases the generalization ability of the model. Applying this loss term, we achieve state-of-the-art results on Penn Treebank (PTB) and WikiText-2 (WT2) language modeling tasks.

[*] Equal contribution. Ordering determined by coin flipping.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 93–100
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64010

## 2 Related Work

Melis et al. (2018) investigated different NLM architectures and regularization methods with the use of a black-box hyperparameter tuner. In particular, the LSTM architecture was compared to two more recent recurrent approaches, namely recurrent highway networks (Zilly et al., 2017) and neural architecture search (Zoph and Le, 2017). They found that the standard LSTM architecture outperforms other models, if properly regularized.

Merity et al. (2017a) used various regularization methods such as activation regularization (Merity et al., 2017b) in a LSTM model. They also introduced a variant of the averaged stochastic gradient method, where the averaging trigger is not tuned by the user but relies on a non-monotonic condition instead. With these and further regularization and optimization methods, improved results on PTB and WT2 were achieved.

To further improve this network architecture, Yang et al. (2018) introduced the mixture of softmaxes (MoS) model, claiming that the calculation of the output probabilities with a single softmax layer is a bottleneck. In their approach, several output probabilities are calculated and then combined via a weighted sum. The LSTM-MoS architecture provides state-of-the-art results on PTB and WT2 at the time of writing and is used as the baseline model for comparisons in this work.

Other works proposed to tie the embedding and projection matrices. Press and Wolf (2017) investigated the effects of weight tying, analyzed update rules after tying and showed that tied matrices evolve in a similar way as the projection matrix. Inan et al. (2017) were motivated by the fact that with a classification setup over the vocabulary, inter-word information is not utilized to its full potential. They also provided theoretical justification on why it is appropriate to tie the above-mentioned matrices.

Besides using the word embedding matrix, there are other approaches to represent word sequences. Zhang et al. (2015) proposed a new embedding method called fixed-sized ordinally-forgetting encoding (FOFE), which allows them to encode variable-length sentences into fixed-length vectors almost uniquely.

Additionally, Salimans and Kingma (2016) introduced a weight normalization reparametrization trick on weight matrices, which separates the norm and the angle of a vector. This can speed up the convergence of stochastic gradient descent and also allows for explicit scaling of gradients in the amplitude and direction. They also discussed the connections between weight normalization and batch normalization.

On top of one-hot representations of words, Irie et al. (2015) used additional information to represent word sequences. It is shown that the use of long-context bag-of-words as additional feature for language modeling can narrow the gap between feed-forward NLMs and recurrent NLMs.

## 3 Neural Language Modeling

In NLM the probability of a word sequence $\boldsymbol{x}_1^t = x_1 x_2 ... x_t$ is decomposed as

$$P(\boldsymbol{x}_1^t) = \prod_{j=1}^{t} P(x_j | \boldsymbol{x}_{j-n+1}^{j-1}) \quad (1)$$

so that the $(n-1)$ preceding words $\boldsymbol{x}_{j-n+1}^{j-1}$ are considered for the prediction of the next word $x_j$. This is typically done by using a recurrent neural network, e.g. a stack of LSTM layers, to encode the input sequence as

$$h_t = \text{LSTM}(E^T [x_{t-n+1}, x_{t-n+2}, ..., x_{t-1}]) \quad (2)$$

where $E^T$ is the transposed embedding matrix, $[x_{t-n+1}, x_{t-n+2}, ..., x_{t-1}]$ are the one-hot encoded preceding words and the LSTM() function returns the last hidden state of the last LSTM layer. The probability distribution over the next word $x_t$ is then calculated as

$$P(x_t = x_k | h_t) = \frac{\exp(W_k h_t)}{\sum_{j=1}^{V} \exp(W_j h_t)} \quad (3)$$

with $V$ being the vocabulary size, $k = 1, 2, ..., V$, and $W_k$ being the $k$-th row vector in the projection matrix $W$.

For training the neural network, the cross-entropy error criterion, which is equivalent to the maximum likelihood criterion, is used. For the $i$-th sequence of words $\boldsymbol{x}_1^{t_i}$, the cross-entropy loss $L_i$ is defined as

$$L_i = -\text{log} P(x_{t_i} = x_{y_i} | h_{t_i}) \quad (4)$$

with $y_i$ being the true label of $x_{t_i}$. The total loss is then calculated as

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i \quad (5)$$

where $N$ is the total number of sequences. A language model is normally scored by perplexity (*ppl*). For a given test corpus $\boldsymbol{x}_1^T = x_1 x_2 ... x_T$, the *ppl* is calculated as

$$ppl = P(\boldsymbol{x}_1^T)^{-\frac{1}{T}} \tag{6}$$

which is a measurement on how likely a given sentence is, according to the prediction of the model.

In the above formulation, we have an embedding matrix $E$ and a projection matrix $W$. When the two matrices are tied and one-hot vectors are used to represent words, the rows of these matrices are then the word vectors of the corresponding words. Particularly, we focus on the norms of the row vectors and study their relationship with word counts and how to regularize them.

## 4 Weight Norm Initialization

We first train models on PTB and WT2 as described in (Yang et al., 2018) and plot the norms of learned weight vectors of the embedding matrix in Figure 1.

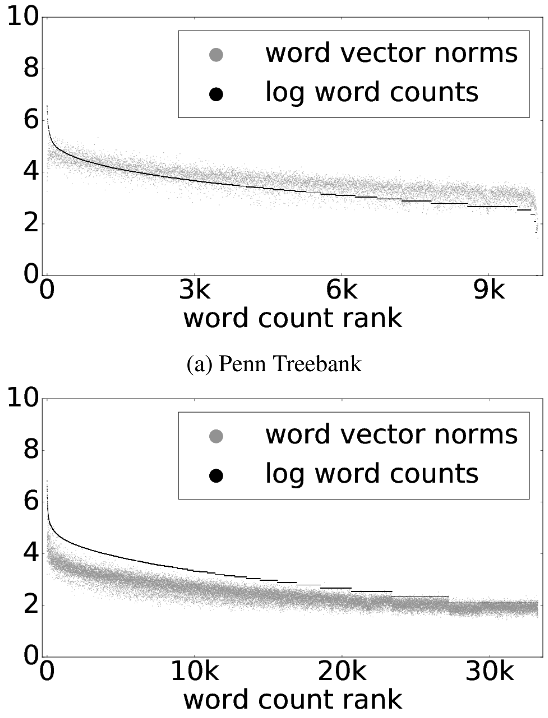
When the words are ranked by their counts and placed on the x-axis from frequent to infrequent, it can be seen that the word vector norms follow a downward trend as well. Log unigram counts are also plotted for comparison. As can be seen, the norm distribution follows a similar trend as the log counts. It is important to note, that the logit for word $x_k$ and context $h_t$ is calculated as $W_k h_t$ (see Equation 3), which can be rewritten as

$$W_k h_t = \|W_k\| \, \|h_t\| \cos(\theta) \tag{7}$$

where $\theta$ denotes the angle between $W_k$ and $h_t$. Therefore, one intuition from the aforementioned observation is that, for a frequent word, the network tends to learn a weight vector $W_k$ with a greater norm to maximize likelihood. This motivates our approach to initialize the weight norms with scaled log counts rather than uniformly random values in a specific range.

Because we wish to initialize the weight norms explicitly with scaled logarithm of the word counts, it is helpful to look at a weight vector's magnitude and direction separately. For this purpose, we use a reparameterization technique on the weight vectors as described in (Salimans and Kingma, 2016):

$$W_k = g_k \frac{v_k}{\|v_k\|_2} \tag{8}$$



(a) Penn Treebank

(b) WikiText-2

Figure 1: Word vector norms of fine-tuned MoS models (Yang et al., 2018), trained on (a) Penn Treebank and (b) WikiText-2. Words are ranked by their counts in a descending order and thus frequent words are to the left. Actual logarithm of word counts are plotted in black, and word vector norms are grey. We observe that word vector norms loosely follow the trend of log counts.

where $k = 1, 2, ..., V$, $g_k = \|W_k\|_2$, and $v_k$ is a vector proportional to $W_k$. Reparameterizing the weight vectors makes it easy to implement the weight norm initialization as

$$g_k = \sigma \log c_k \tag{9}$$

where $c_k$ denotes unigram word count for word $k$ and $\sigma$ is a scalar applied to the log counts. We sample each component of $v_k$ from a continuous uniform distribution in $[-r, r]$, where $r$ is a hyperparameter, specifying the initialization range. With this, no constraint on the weight vector direction is imposed during initialization.

Additionally, we adopt an adaptive gradient strategy which regularizes the gradients in $g_k$. As in

$$(\frac{\partial L}{\partial g})' = \begin{cases} [1-(1-\gamma)\frac{t}{\tau}]\frac{\partial L}{\partial g}, & \text{for } t \le \tau \\ \gamma \frac{\partial L}{\partial g}, & \text{for } t > \tau \end{cases} \tag{10}$$

when epoch $t$ is no greater than a specified epoch

| | | Tokens | Vocab Size |
|---|---|---|---|
| **Penn Treebank** | Train | 888k | 10k |
| | Valid | 70k | |
| | Test | 79k | |
| **WikiText-2** | Train | 2.1M | 33k |
| | Valid | 214k | |
| | Test | 241k | |

Table 1: Statistics of the Penn Treebank and WikiText-2 datasets.

$\tau$, $(\frac{\partial L}{\partial g_k})'$ — the regularized gradient in $g_k$, linearly decays to $\gamma$ ($\gamma \leq 1$) times the unregularized gradient $\frac{\partial L}{\partial g_k}$. Otherwise, we directly use the discounted gradient. In analogy to learning rate decay, this adaptive gradient strategy anneals the word vector norm updates in each step. The intuition for such a strategy is that after a certain amount of epochs, the weight norms should not change so drastically from the initialized scaled log counts.

## 5 Weight Norm Regularization

Weight regularization (WR) is a well established method to combat overfitting in neural networks, which is especially important on smaller datasets (Krogh and Hertz, 1992). The idea is to push weights in the network to zero, where gradients are not significant. Typically, WR is implemented by adding an extra term to the loss function $L_0$, which penalizes the norm of all weights in the network. For example, L$_2$-regularization is implemented as

$$L = L_0 + \frac{\lambda}{2}\sum_{w}(\|w\|_2)^2 \qquad (11)$$

with the sum going over all weights $w$ in the network and $\lambda$ being the regularization strength. However, this method is not perfect, as it affects every weight in the network equally and may lead to hidden units' weights getting stuck near zero.

In this work we add a constraint specifically on the embedding and projection matrices, whose weights are shared. Since the row vectors in both matrices are word vectors, it seems appropriate to put constraints explicitly on their norms instead of on each individual weight parameter in the matrices.

We propose to add a regularization term to the standard loss function $L_0$ in the form of

$$L_{wr} = L_0 + \rho\sqrt{\sum_{j=1}^{V}(\|W_j\|_2 - \nu)^2} \qquad (12)$$

where $\nu$, $\rho \geq 0$ are two scalars and $W_j$ is the $j$-th row vector of the projection matrix $W$. The L$_2$-norms of the row vectors are pushed towards $\nu$, while $\rho$ is the regularization strength. This will punish the row vectors for adopting norms other than $\nu$, in the hope of reducing the effect of overfitting on the training data.

The choice of a soft regularization loss term instead of hard-fixing the weight norms in the forward pass is motivated by the weight norm distribution shown in Figure 1. It can be seen that NLMs tend to learn non-equal weight norms for words with different counts. Therefore, hard-fixing weight norms may limit the network's ability to learn.

## 6 Experiments

### 6.1 Experiment Setup

The experiments are conducted on two popular language modeling datasets. The number of tokens and size of vocabulary for each dataset are summarized in Table 1.

| epoch | Penn Treebank | | | WikiText-2 | | |
|---|---|---|---|---|---|---|
| | wni *ppl* | baseline *ppl* | *ppl* reduction (%) | wni *ppl* | baseline *ppl* | *ppl* reduction (%) |
| 1 | 162.18 | 180.72 | 10.26 | 172.19 | 192.19 | 10.41 |
| 10 | 85.92 | 92.09 | 6.70 | 95.90 | 100.72 | 4.79 |
| 20 | 73.36 | 78.94 | 7.07 | 85.14 | 88.21 | 3.48 |
| 30 | 71.44 | 73.06 | 2.22 | 81.80 | 82.70 | 1.09 |
| 40 | 69.27 | 70.20 | 1.32 | 79.28 | 80.32 | 1.29 |

Table 2: Perplexity (*ppl*) improvement using weight norm initialization (wni) in early epochs on Penn Treebank and WikiText-2. *ppl* reduction is around 10% after the first epoch on both tasks, and decays to approximately 1% after 40 epochs. The wni model has slightly higher perplexities than the baseline model from around 50 epochs onward.

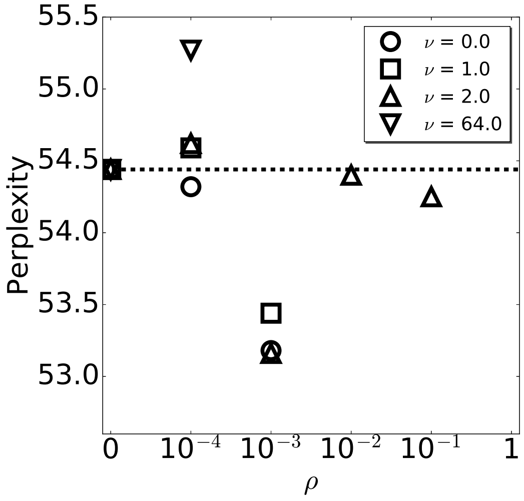
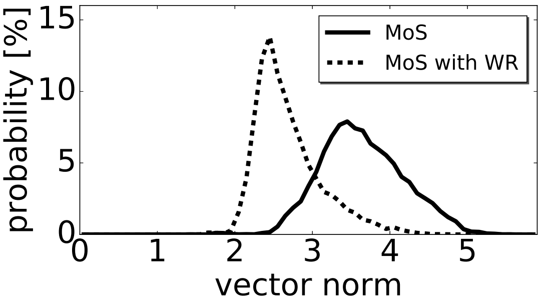
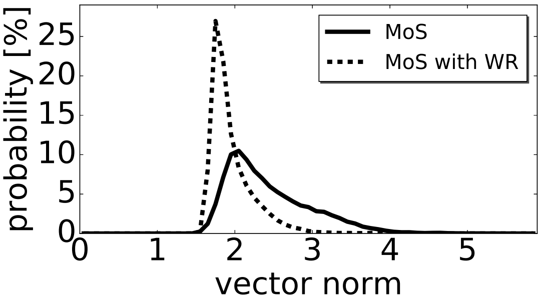


Figure 2: Model perplexity on the Penn Treebank test set as a function of $\rho$. The different symbols denote different values of $\nu$. Models not depicted yield higher perplexity values. The doted line marks the baseline result (with $\rho = 0$) as reported by Yang et al. (2018).



(a) Penn Treebank



(b) WikiText-2

Figure 3: Weight norm distributions of the projection matrices' row vectors for the AWD-LSTM-MoS model from Yang et al. (2018) as well as for our regularized version (WR). The models are trained on the (a) Penn Treebank corpus and (b) WikiText-2 corpus with the resulting test perplexities shown in Table 3 and Table 4 respectively.

The smaller one is the PTB corpus with preprocessing from Mikolov et al. (2010), which has a comparatively small vocabulary size of 10k. With a smaller number of sentences, this dataset is a good choice for performing optimization of hyperparameters. The second corpus WT2, which was introduced by Merity et al. (2016), has over three times the vocabulary size of PTB.

We use the network structure introduced by Yang et al. (2018) with the same hyper-parameter values to ensure comparability. Several regularization techniques are used in this setup, such as dropout and weight decay. Furthermore, the embedding and projection matrices are tied by default. For optimization, we adopt the same strategy as described in (Merity et al., 2017a). That is, a conservative non-monotonic criterion is used to switch from stochastic gradient descent (SGD) to averaged stochastic gradient descent (ASGD) (Polyak and Juditsky, 1992). For more details of the network structure refer to (Yang et al., 2018).

## 6.2 Weight Norm Initialization

We tune the hyperparameter $\sigma$ and use a value of $\sigma = 0.5$ to scale the logarithm of word counts. Initialization range $r$ is set to 0.1 for both the reparametrized direction vectors and the baseline word vectors. Empirically, we set $\gamma = 0.1$ and $\tau = 100$ for the adaptive gradient method. Perplexities on both PTB and WT2 in early epochs, as well as the relative perplexity improvement over baseline models are summarized in Table 2.

First, we notice significant improvement after the first epoch of training using weight norm initialization. About 10% of perplexity reduction is achieved on both datasets. This could be beneficial, when one wants to train on large datasets and/or can only train for a limited number of epochs. Second, the perplexity improvements decay down to around 1% after 40 epochs. This is in agreement with our expectation, because apart from reduced gradient in $g_k$, a weight norm initialized model is not fundamentally different from the baseline model and no major difference should be seen if we train for long enough. It is important to note that with only weight norm initialization, both models eventually converge to perplexities that are slightly worse than the baseline. We also notice that the epochs, after which the optimizer is switched from SGD to ASGD, are different in weight norm initialized models and baseline models.

| **Model** | **#Params** | **Validation** | **Test** |
|---|---|---|---|
| Mikolov and Zweig (2012) - RNN-LDA + KN + cache | 9M | - | 92.0 |
| Zaremba et al. (2014) - LSTM | 20M | 86.2 | 82.7 |
| Gal and Ghahramani (2016) - Variational LSTM (MC) | 20M | - | 78.6 |
| Kim et al. (2016) - CharCNN | 19M | - | 78.9 |
| Merity et al. (2016) - Pointer Sentinel-LSTM | 21M | 72.4 | 70.9 |
| Grave et al. (2017) - LSTM + continuous cache pointer$^\dagger$ | - | - | 72.1 |
| Inan et al. (2017) - Tied Variational LSTM + augmented loss | 24M | 75.7 | 73.2 |
| Zilly et al. (2017) - Variational RHN | 24M | 75.7 | 73.2 |
| Zoph and Le (2017) - NAS Cell | 25M | - | 64.0 |
| Melis et al. (2018) - 2-layer skip connection LSTM | 24M | 60.9 | 58.3 |
| Merity et al. (2017a) - AWD-LSTM | 24M | 60.0 | 57.3 |
| Yang et al. (2018) - AWD-LSTM-MoS | 22M | 56.54 | 54.44 |
| Ours - AWD-LSTM-MoS with weight norm regularization | 22M | **55.03** | **53.16** |

Table 3: Single model perplexity on the Penn Treebank test and validation sets. Baseline results are obtained from (Yang et al., 2018). † indicates the use of dynamic evaluation.

| **Model** | **#Params** | **Validation** | **Test** |
|---|---|---|---|
| Inan et al. (2017) - Variational LSTM + augmented loss | 28M | 91.5 | 87.0 |
| Grave et al. (2017) - LSTM + continuous cache pointer$^\dagger$ | - | - | 68.9 |
| Melis et al. (2018) - 2-layer skip connection LSTM | 24M | 69.1 | 65.9 |
| Merity et al. (2017a) - AWD-LSTM | 33M | 69.1 | 66.0 |
| Yang et al. (2018) - AWD-LSTM-MoS | 35M | 63.88 | 61.45 |
| Ours - AWD-LSTM-MoS with weight norm regularization | 35M | **62.67** | **60.13** |

Table 4: Single model perplexity on the WikiText-2 test and validation sets. Baseline results are obtained from (Yang et al., 2018). † indicates the use of dynamic evaluation.

### 6.3 Weight Norm Regularization

In order to tune the hyperparameters $\rho$ and $\nu$ introduced in Section 5, we perform a grid search over the PTB dataset, the results of which are shown in Figure 2. If the norm constraint $\nu$ becomes too large, perplexity worsens significantly, as seen in the case of $\nu = 64$. A model with a $\nu$-value of 2 provides the best result in most cases. We hypothesize that a value of $\nu$ that is too small results in the logit being close to zero as shown in Equation 7. For the regularization strength $\rho$, we recognize that $\rho = 10^{-3}$ gives the best result on the PTB test data. Larger or smaller values can hurt the performance of the system, depending also on the value of $\nu$. It should be noted that the optimized value of $\rho$ is significantly larger than the scaling $s_{wd}$ of the weight decay term, which was optimized to be $1.2\times10^{-6}$ by Merity et al. (2017a).

The resulting weight norm distributions of the projection matrices' row vectors are shown in Figure 3a and Figure 3b for models trained on PTB and WT2 respectively. Our efforts of pushing the norms to a value of $\nu = 2.0$ resulted in a noticeably smaller average norm, as well as in a overall more narrow distribution.

With the tuned parameter values $\rho = 10^{-3}$ and $\nu = 2.0$ we improve the previous state-of-the-art result by 1.28 *ppl* on PTB and by 1.32 *ppl* on WT2 (without considering dynamic evaluation (Krause et al., 2018), see Table 3 and Table 4). This is achieved without increasing the number of trainable parameters in the network or slowing down the training process.

## 7 Conclusion

Word embedding matrix and output projection matrix are important components in LSTM-based LMs. They are also widely used in other NLP models where one-hot vectors of words need to be mapped into lower dimensional space. Given the one-hot nature of word representations, row vectors in such matrices are then the corresponding word vectors. We study specifically the norms of these learned word vectors, the distribution of the norms, and the relationship with word counts. We show that with a simple initialization strategy together with a reparametrization technique, it is possible to get significantly lower perplexity in early epochs during training. By using a weight norm regularization loss term, we are able to obtain significant improvements on standard language modeling tasks — 2.4% *ppl* reduction on PTB and 2.1% on WT2.

We propose three directions to investigate further. First, in this work we use scaled logarithm of word counts to initialize the weight norms. It is a logical next step to use smoothing techniques on the word counts and study the effects of such initializations. Second, we currently apply the same norm constraint on different words. Altering the loss function and regularizing the weight norms to word counts (and smoothed word counts) is worth examining as well. Finally, our focus so far is on weight norms. It is a more exciting and challenging task to study the pairwise inner products, and single out the effects of angular differences.

We also plan to expand our regularization and initialization techniques to the field of neural machine translation. Embedding and projection matrices are also present in neural machine translation networks, which could potentially benefit from our methods as well. It seems natural to use our methods on the transformer architecture introduced by Vaswani et al. (2017), in which the embedding matrices at source and target sides, plus the projection matrix, are three-way tied.

## Acknowledgments

This work has received funding from the European Research Council (ERC) (under the European Union's Horizon 2020 research and innovation programme, grant agreement No 694537, project "SEQCLAS") and the Deutsche Forschungsgemeinschaft (DFG; grant agreement NE 572/8-1, project "CoreTec"). The GPU computing cluster was supported by DFG (Deutsche Forschungsgemeinschaft) under grant INST 222/1168-1 FUGG.

The work reflects only the authors' views and none of the funding agencies is responsible for any use that may be made of the information it contains.

## References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. In *Journal of machine learning research*, volume 3, pages 1137–1155.

A. Corazza, R. De Mori, R. Gretter, R. Kuhn, and G. Satta. 1995. Language models for automatic speech recognition. In *Speech Recognition and Coding*, pages 157–173. Springer.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017. Improving neural language models with a continuous cache. In *International Conference on Learning Representations*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. In *International Conference on Learning Representations*.

Kazuki Irie, Ralf Schlüter, and Hermann Ney. 2015. Bag-of-words input for long history representation in neural network-based language models for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Slava Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE transactions on acoustics, speech, and signal processing*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*, pages 2741–2749.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, pages 181–184.

Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. 2018. Dynamic evaluation of neural sequence models. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2766–2775.

Anders Krogh and John A. Hertz. 1992. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957.

Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *International Conference on Learning Representations*.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017a. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*.

Stephen Merity, Bryan McCann, and Richard Socher. 2017b. Revisiting activation regularization for language rnns. *arXiv preprint arXiv:1708.01009*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomas Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *11th Annual Conference of the International Speech Communication Association*.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *IEEE Spoken Language Technology Workshop*, pages 234–239.

Jan-Thorsten Peter, Andreas Guta, Tamer Alkhouli, Parnia Bahar, Jan Rosendahl, Nick Rossenbach, Miguel Graça, and Hermann Ney. 2017. The RWTH Aachen University english-german and german-english machine translation system for WMT 2017. In *Proceedings of the Second Conference on Machine Translation*, pages 358–365.

Boris T. Polyak and Anatoli B. Juditsky. 1992. Acceleration of stochastic approximation by averaging. In *SIAM Journal on Control and Optimization*, pages 838–855.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 157–163.

Tim Salimans and Diederik P Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech & Language*, 21(3):492–518.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank rnn language model. In *International Conference on Learning Representations*.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Shiliang Zhang, Hui Jiang, Mingbin Xu, Junfeng Hou, and Lirong Dai. 2015. The fixed-size ordinally-forgetting encoding method for neural network language models. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 495–500.

Julian Georg Zilly, Rupesh Kumar Srivastava, Jan Koutník, and Jürgen Schmidhuber. 2017. Recurrent highway networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 4189–4198.

Barret Zoph and Quoc V Le. 2017. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*.

# Contextual Neural Model for Translating Bilingual Multi-Speaker Conversations

**Sameen Maruf**[1]
Monash University
VIC, Australia

**André F. T. Martins**[2]
Unbabel &
Instituto de Telecomunicacões
Lisbon, Portugal

**Gholamreza Haffari**[1]
Monash University
VIC, Australia

[1]{firstname.lastname}@monash.edu
[2]andre.martins@unbabel.com

## Abstract

Recent works in neural machine translation have begun to explore document translation. However, translating online multi-speaker conversations is still an open problem. In this work, we propose the task of translating Bilingual Multi-Speaker Conversations, and explore neural architectures which exploit both source and target-side conversation histories for this task. To initiate an evaluation for this task, we introduce datasets extracted from Europarl v7 and OpenSubtitles2016. Our experiments on four language-pairs confirm the significance of leveraging conversation history, both in terms of BLEU and manual evaluation.

## 1 Introduction

Translating a conversation online is ubiquitous in real life, e.g. in the European Parliament, United Nations, and customer service chats. This scenario involves leveraging the conversation history in multiple languages. The goal of this paper is to propose and explore a simplified version of such a setting, referred to as Bilingual Multi-Speaker Machine Translation (Bi-MSMT), where speakers' turns in the conversation switch the source and target languages. We investigate neural architectures that exploit the bilingual conversation history for this scenario, which is a challenging problem as the history consists of utterances in both languages.

The ultimate aim of all machine translation systems for dialogue is to enable a multi-lingual conversation between multiple speakers. However, translation of such conversations is not well-explored in the literature. Recently, there has been work focusing on using the discourse or document context to improve NMT, in an online setting, by using the past context (Jean et al., 2017; Wang et al., 2017; Bawden et al., 2017; Voita et al., 2018), and in an offline setting, using the past and future context (Maruf and Haffari, 2018). In this paper, we design and evaluate a conversational Bi-MSMT model, where we incorporate the source and target-side conversation histories into a sentence-based attentional model (Bahdanau et al., 2015). Here, the source history comprises of sentences in the original language for both languages, and the target history consists of their corresponding translations. We experiment with different ways of computing the source context representation for this task. Furthermore, we present an effective approach to leverage the target-side context, and also present an intuitive approach for incorporating both contexts simultaneously. To evaluate this task, we introduce datasets extracted from Europarl v7 and OpenSubtitles2016, containing speaker information. Our experiments on English-French, English-Estonian, English-German and English-Russian language-pairs show improvements of +1.44, +1.16, +1.75 and +0.30 BLEU, respectively, for our best model over the context-free baseline. The results show the impact of conversation history on translation of bilingual multi-speaker conversations and can be used as benchmark for future work on this task.

## 2 Related Work

Our research builds upon prior work in the field of context-based language modelling and context-based machine translation.

**Language Modelling** There have been few works on leveraging context information for language modelling. Ji et al. (2015) introduced Document Context Language Model (DCLM) which incorporates inter and intra-sentential contexts. Hoang et al. (2016) make use of side information, e.g. metadata, and Tran et al. (2016) use inter-document context to boost the performance

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 101–112
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64011

of RNN language models.

For conversational language modelling, Ji and Bilmes (2004) propose a statistical multi-speaker language model (MSLM) that considers words from other speakers when predicting words from the current one. By taking the inter-speaker dependency into account using a normal trigram context, they report significant reduction in perplexity.

**Statistical Machine Translation** The few SMT-based attempts to document MT are either restrictive or do not lead to significant improvements upon automatic evaluation. Few of these deal with specific discourse phenomena, such as resolving anaphoric pronouns (Hardmeier and Federico, 2010) or lexical consistency of translations (Garcia et al., 2017). Others are based on a two-pass approach i.e., to improve the translations already obtained by a sentence-level model (Hardmeier et al., 2012; Garcia et al., 2014).

**Neural Machine Translation** Using context-based neural models for improving online and offline NMT is a popular trend recently. Jean et al. (2017) extend the vanilla attention-based NMT model (Bahdanau et al., 2015) by conditioning the decoder on the previous source sentence via a separate encoder and attention component. Wang et al. (2017) generate a summary of three previous source sentences via a hierarchical RNN, which is then added as an auxiliary input to the decoder. Bawden et al. (2017) explore various ways to exploit context from the previous sentence on the source and target-side by extending the models proposed by Jean et al. (2017); Wang et al. (2017). Apart from being difficult to scale, they report deteriorated BLEU scores when using the target-side context.

Tu et al. (2017) augment the vanilla NMT model with a continuous cache-like memory, along the same lines as the cache-based system for traditional document MT (Gong et al., 2011), which stores hidden representations of recently generated words as translation history. The proposed approach shows significant improvements over all baselines when translating subtitles and comparable performance for news and TED talks. Along similar lines, Kuang et al. (2018) propose dynamic and topic caches to capture contextual information either from recently translated sentences or the entire document to model coherence for NMT. Voita et al. (2018) introduce a context-aware NMT model in which they control and analyse the flow of information from the extended context to the translation model. They show that using the previous sentence as context their model is able to implicitly capture anaphora.

For the offline setting, Maruf and Haffari (2018) incorporate the global source and target document contexts into the base NMT model via memory networks. They report significant improvements using BLEU and METEOR for the contextual model over the baseline. To the best of our knowledge, there has been no work on Multi-Speaker MT or its variation to date.

## 3 Preliminaries

### 3.1 Problem Formulation

We are given a dataset that comprises parallel conversations, and each conversation consists of *turns*. Each turn is constituted by sentences spoken by a single speaker, denoted by $\mathbf{x}$ or $\mathbf{y}$, if the sentence is in English or Foreign language, respectively. The goal is to learn a model that is able to leverage the mixed-language conversation history in order to produce high quality translations.

### 3.2 Data

Standard machine translation datasets are inappropriate for Bi-MSMT task since they are not composed of conversations or the speaker annotations are missing. In this section, we describe how we extract data from raw Europarl v7 (Koehn, 2005) and OpenSubtitles2016[1] (Lison and Tiedemann, 2016) for this task[2].

**Europarl** The raw Europarl v7 corpus (Koehn, 2005) contains `SPEAKER` and `LANGUAGE` tags where the latter indicates the language the speaker was actually using. The individual files are first split into conversations. The data is tokenised (using scripts by Koehn (2005)), and cleaned (headings and single token sentences removed). Conversations are divided into smaller ones if the number of speakers is greater than 5.[3] The corpus is then randomly split into train/dev/test sets with respect to conversations in ratio 100:2:3. The English side of the corpus is set as reference, and

[1] http://www.opensubtitles.org/

[2] The data is publicly available at https://github.com/sameenmaruf/Bi-MSMT.git

[3] Using the conversations as is or setting a higher threshold further reduces the data due to inconsistencies in conversation/turn lengths in the source and target side.

|  | **Europarl** |  |  | **Subtitles** |
|---|---|---|---|---|
|  | **En-Fr** | **En-Et** | **En-De** | **En-Ru** |
| # Conversations | 6997 | 4394 | 3582 | 23126 |
| # Sentences | 246540 | 174218 | 109241 | 291516 |
| **Mean Statistics per Conversation** |  |  |  |  |
| # Sentences | 36.24 | 40.65 | 31.50 | 13.60 |
| # Turns | 4.77 | 4.85 | 4.79 | 7.12 |
| Turn Length | 7.12 | 7.92 | 6.16 | 1.68 |

Table 1: General statistics for training set.

if the language tag is absent, the source language is English, otherwise Foreign. The sentences in the source-side of the corpus are kept or swapped with those in the target-side based on this tag.

We perform the aforementioned steps for English-French, English-Estonian and English-German, and obtain the bilingual multi-speaker corpora for the three language pairs. Before splitting into train/dev/test sets, we remove conversations with sentences having more than 100 tokens for English-French, English-German and more than 80 tokens for English-Estonian[4] respectively, to limit the sentence-length for using subwords with BPE (Sennrich et al., 2016). The data statistics are given in Table 1 and Appendix A[5].

**Subtitles** There has been recent work to obtain speaker labels via automatic turn segmentation for the OpenSubtitles2016 corpus (Lison and Meena, 2016; van der Wees et al., 2016; Wang et al., 2016). We obtain the English side of OpenSubtitles2016 corpus annotated with speaker information by Lison and Meena (2016).[6] To obtain the parallel corpus, we use the OpenSubtitles alignment links to align foreign subtitles to the annotated English ones. For each subtitle, we extract individual conversations with more than 5 sentences and at least two turns. Conversations with more than 30 turns are discarded. Finally, since subtitles are in a single language, we assign language tag such that the same language occurs in alternating turns. We thus obtain the Bi-MSMT corpus for English-Russian, which is then divided into training, development and test sets.

### 3.3 Sentence-based attentional model

Our base model consists of two sentence-based NMT architectures (Bahdanau et al., 2015), one for each translation direction. Each of them contains an encoder to *read* the source sentence and an attentional decoder to *generate* the target translation one token at a time.

**Encoder** It maps each source word $x_m$ to a distributed representation $\boldsymbol{h}_m$ which is the concatenation of the corresponding hidden states of two RNNs running in opposite directions over the source sentence. The forward and backward RNNs are taken to be GRUs (gated-recurrent unit; Cho et al. (2014)) in this work.

**Decoder** The generation of each target word $y_n$ is conditioned on all the previously generated words $\boldsymbol{y}_{<n}$ via the state $\boldsymbol{s}_n$ of the decoder, and the source sentence via a *dynamic* context vector $\boldsymbol{c}_n$:

$$\begin{aligned} y_n &\sim \text{softmax}(\boldsymbol{W}_y \cdot \boldsymbol{u}_n + \boldsymbol{b}_y) \\ \boldsymbol{u}_n &= \tanh(\boldsymbol{s}_n + \boldsymbol{W}_{uc} \cdot \boldsymbol{c}_n + \boldsymbol{W}_{un} \cdot \boldsymbol{E}_T[y_{n-1}]) \\ \boldsymbol{s}_n &= \text{GRU}(\boldsymbol{s}_{n-1}, \boldsymbol{E}_T[y_{n-1}], \boldsymbol{c}_n) \end{aligned}$$

where $\boldsymbol{E}_T[y_{n-1}]$ is the embedding of previous target word $y_{n-1}$, and $\{\boldsymbol{W}_{(\cdot)}, \boldsymbol{b}_y\}$ are the parameters. The fixed-length *dynamic* context representation of the source sentence $\boldsymbol{c}_n = \sum_m \alpha_{nm} \boldsymbol{h}_m$ is generated by an attention mechanism where $\boldsymbol{\alpha}$ specifies the proportion of relevant information from each word in the source sentence.

## 4 Conversational Bi-MSMT Model

Before we delve into the details of how to leverage the conversation history, we identify the three types of context we may encounter in an ongoing bilingual multi-speaker conversation, as shown in Figure 1. It comprises of: (i) the previously completed English turns, (ii) the previously completed Foreign turns, and (iii) the ongoing turn (English or Foreign).

We propose a conversational Bi-MSMT model that is able to incorporate all three types of context using source, target or dual conversation histories into the base model. The base model caters to the speaker's language transition by having one sentence-based NMT model (described previously) for each translation direction, English→Foreign and Foreign→English. We now

[4] Sentence-lengths of 100 tokens result in longer sentences than what we get for the other two language-pairs.

[5] Although the extracted dataset is small but we believe it to be a realistic setting for a real-world conversation task, where reference translations are usually not readily available and expensive to obtain.

[6] The majority of sentences still have missing annotations (Lison and Meena, 2016) due to changes between the original script and the actual movie or alignment problems between scripts and subtitles. As for Wang et al. (2016), their publicly released data is even smaller than our En-De dataset extracted from Europarl.

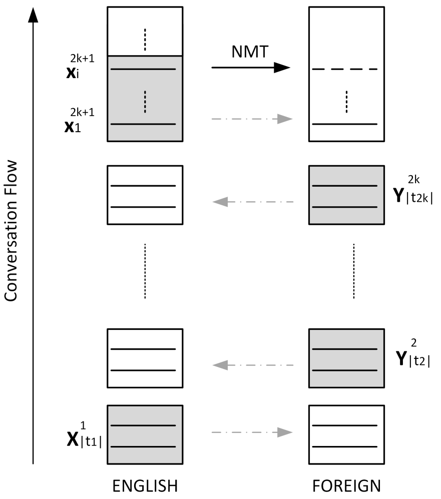


Figure 1: Overview of an ongoing conversation while translating $i^{th}$ sentence in $2k+1^{th}$ turn. $\mathbf{X}^j_{|t_j|}$ and $\mathbf{Y}^j_{|t_j|}$ denote the sentences in previous English and Foreign turn respectively, and $\mathbf{x}^j_i$ denotes the sentence $i$ in ongoing turn $j$ where $i \in \{1, ..., |t_j|\}$. The shaded turns are observed i.e., source (the speaker utterances), while the rest are unobserved i.e., the target translations or the unuttered source sentences for current turn.

describe our approach for extracting relevant information from the source and target bilingual conversation history.

### 4.1 Source-Side History

Suppose we are translating an ongoing conversation having alternating turns of English and Foreign. We are currently in the $2k+1^{th}$ turn (in English) and want to translate its $i^{th}$ sentence using the source-side conversation history represented by context vector $\mathbf{o}_{src}$ (dimensions $H$).

Let's assume that we already have the representations of previous source sentences in the conversation. We pass the source sentence representations through Turn-RNNs, which are composed of language-specific bidirectional RNNs irrespective of the speaker, as shown in Figure 2, and concatenate the last hidden states of the forward and backward Turn-RNNs to get the final turn representation $\mathbf{r}_j$, where $j$ denotes the turn index. The individual turn representations are then combined, based on language[7], to obtain context vectors $\mathbf{o}_{en}$ and $\mathbf{o}_{fr}$, computed in several possible ways (described below), which are further amalgamated using a gating mechanism so as to give differing importance to each element of the context vector:

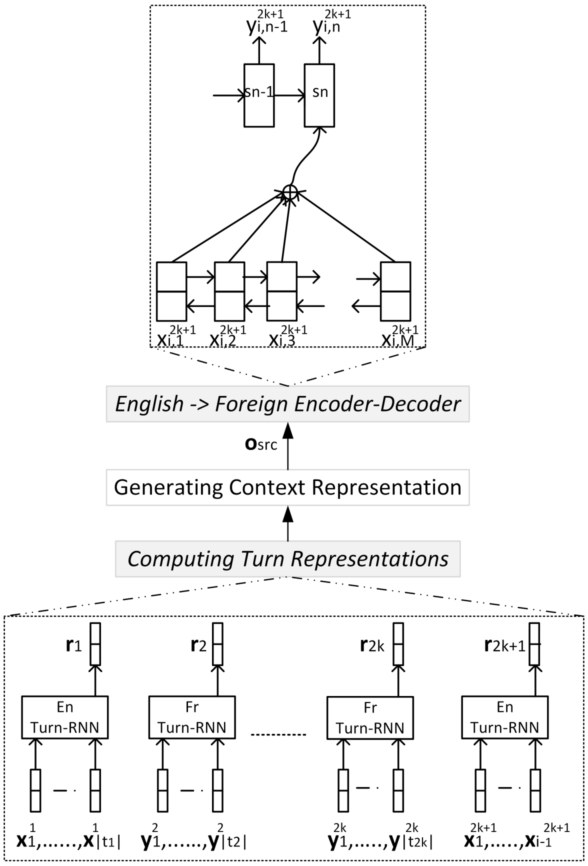

$y^{2k+1}_{i,n-1}$
$y^{2k+1}_{i,n}$
Sn-1
Sn
$x^{2k+1}_{i,1}$
$x^{2k+1}_{i,2}$
$x^{2k+1}_{i,3}$
$x^{2k+1}_{i,M}$
English -> Foreign Encoder-Decoder
$o_{src}$
Generating Context Representation
Computing Turn Representations
r1
r2
r2k
r2k+1
En Turn-RNN
Fr Turn-RNN
Fr Turn-RNN
En Turn-RNN
$x^1_1,......,x^1_{|t1|}$
$y^2_1,......,y^2_{|t2|}$
$y^{2k}_1,.....,y^{2k}_{|t2k|}$
$x^{2k+1}_1,.....,x^{2k+1}_{i-1}$

Figure 2: Architectural overview when translating $i^{th}$ sentence in $2k+1^{th}$ turn using source history.

$$\begin{aligned} \mathbf{o}_{en,fr} &= \boldsymbol{\alpha} \odot \mathbf{o}_{en} + (\mathbf{1} - \boldsymbol{\alpha}) \odot \mathbf{o}_{fr} \qquad (1) \\ \boldsymbol{\alpha} &= \sigma(\mathbf{U}_{en} \times \mathbf{o}_{en} + \mathbf{U}_{fr} \times \mathbf{o}_{fr} + \mathbf{b}_g) \end{aligned}$$

where $\sigma$ is the logistic sigmoid function, $\mathbf{U}$'s are matrices and $\mathbf{b}_g$ is a vector. Finally, we perform a dimensionality reduction to obtain:

$$\mathbf{o}_{src} = \tanh(\mathbf{W}_T \times \mathbf{o}_{en,fr} + \mathbf{b}_T) \qquad (2)$$

In the remainder of this section, $\{\mathbf{W}, \mathbf{U}, \mathbf{b}\}$ are language-specific learned parameters. We propose five ways of computing the language-specific context representations, $\mathbf{o}_{en}$ and $\mathbf{o}_{fr}$.

**Direct Transformation** The simplest approach is to combine turn representations using a language-specific dimensionality reduction transformation:

$$\begin{aligned} \mathbf{o}_{en} &= \tanh([\mathbf{W}_{en}; ...; \mathbf{W}_{en}] \times [\mathbf{r}_1; ...; \mathbf{r}_{2k+1}] + \mathbf{b}_{en}) \\ \mathbf{o}_{fr} &= \tanh([\mathbf{W}_{fr}; ...; \mathbf{W}_{fr}] \times [\mathbf{r}_2; ...; \mathbf{r}_{2k}] + \mathbf{b}_{fr}) \end{aligned}$$

Here $\mathbf{r}_j$'s are concatenated row-wise.

[7] For this work, we define the turns based on language and do not use the speaker information as for real-world chat scenarios (e.g., agent-client in a customer service chat), we do not have multiple speakers based on language. We leave this for future exploration.

**Hierarchical Gating** We propose a language-specific exponential decay gating based on the intuition that the farther the previous turns are from the current one, the lesser their impact may be on the translation of a sentence in an ongoing turn, similar in spirit to the caching mechanism by Tu et al. (2017):

$$\mathbf{o}_{en} = g_{en}(g_{en}(...g_{en}(g_{en}(\mathbf{r}_1, \mathbf{r}_3), \mathbf{r}_5)...), \mathbf{r}_{2k-1}), \mathbf{r}_{2k+1})$$

where

$$\begin{aligned} g_{en}(\mathbf{a}, \mathbf{b}) &= \boldsymbol{\alpha} \odot \mathbf{a} + (\mathbf{1} - \boldsymbol{\alpha}) \odot \mathbf{b} \\ \boldsymbol{\alpha} &= \sigma(\mathbf{U}_{1,en} \times \mathbf{a} + \mathbf{U}_{2,en} \times \mathbf{b} + \mathbf{b}_{en}) \end{aligned}$$

$\mathbf{o}_{fr}$ is computed in a similar way.

**Language-Specific Attention** The English and Foreign turn representations are combined separately via attention to allow the model to focus on relevant turns in the English and the Foreign context:

$$\begin{aligned} \mathbf{p}_{en} &= \text{softmax}([\mathbf{r}_1; ...; \mathbf{r}_{2k+1}]^T \times \mathbf{h}_i) \qquad (3) \\ \mathbf{p}_{fr} &= \text{softmax}([\mathbf{r}_2; ...; \mathbf{r}_{2k}]^T \times \tanh(\mathbf{W}_{en} \times \mathbf{h}_i + \mathbf{b}_{en})) \\ \mathbf{o}_{en} &= \tanh(\mathbf{W}_{en} \times ([\mathbf{r}_1; ...; \mathbf{r}_{2k+1}] \times \mathbf{p}_{en}) + \mathbf{b}_{en}) \\ \mathbf{o}_{fr} &= [\mathbf{r}_2; ...; \mathbf{r}_{2k}] \times \mathbf{p}_{fr} \end{aligned}$$

Here $\mathbf{r}_j$'s are concatenated column-wise, $\mathbf{h}_i$ is the concatenation of last hidden state of forward and backward RNNs in the encoder for current sentence $i$ in turn $2k+1$ (dimensions *2H*) and $\{\mathbf{W}_{en}$, $\mathbf{b}_{en}\}$ transform the language space to that of the target language.

**Combined Attention** This is a language-independent attention that merges all turn representations into one. The hypothesis here is to verify if the model actually benefits from Language-Specific attention or not.

$$\begin{aligned} \mathbf{p}_{en,fr} &= \text{softmax}([\mathbf{r}_{1,en}; \mathbf{r}_2; ...; \mathbf{r}_{2k+1,en}]^T \times \\ &\qquad \tanh(\mathbf{W}_{en} \times \mathbf{h}_i + \mathbf{b}_{en})) \\ \mathbf{o}_{en,fr} &= [\mathbf{r}_{1,en}; \mathbf{r}_2; ...; \mathbf{r}_{2k+1,en}] \times \mathbf{p}_{en,fr} \end{aligned}$$

Here $\mathbf{r}_{2k+1,en} = \tanh(\mathbf{W}_{en} \times \mathbf{r}_{2k+1} + \mathbf{b}_{en})$.

**Language-Specific Sentence-level Attention** All the previous approaches for computing $\mathbf{o}_{en}$ and $\mathbf{o}_{fr}$ use a single turn-level representation. We propose to use the sentence information explicitly via a sentence-level attention to evaluate the significance of more fine-grained context in contrast to Language-Specific Attention. We first concatenate the hidden states of forward and backward Turn-RNNs for each sentence and get a matrix comprising of representations of all the previous source sentences, i.e., for English turns, we have $[\mathbf{r}_1^1; ...; \mathbf{r}_{|t_1|}^1; ...; \mathbf{r}_1^{2k+1}; ...; \mathbf{r}_{i-1}^{2k+1}]$, and similarly we have another matrix for all the previous Foreign sentences. Here, each $\mathbf{r}_i^j$ is the representation of source sentence $i$ in turn $j$ computed by the bidirectional Turn-RNN. The remaining computations are same as in Eq. 3.

## 4.2 Target-Side History

Using target-side conversation history is as important as that of the source-side since it helps in making the translation more faithful to the target language. This becomes crucial for translating conversations where the previous turns are all in the same language. For incorporating the target-side context, we use a sentence-level attention similar to the one described for the source-side context, i.e., for all previous English source sentences, we have a matrix $\mathbf{R}_{en}$ comprising of the corresponding target sentence representations in Foreign, and another matrix $\mathbf{R}_{fr}$ of target sentence representations (in English) for previous Foreign turns. Here each target sentence representation has dimensions *H*. Then,

$$\begin{aligned} \mathbf{p}_{en} &= \text{softmax}(\mathbf{R}_{en}^T \times \tanh(\mathbf{W}_{t,en} \times \mathbf{h}_i + \mathbf{b}_{t,en})) \\ \mathbf{p}_{fr} &= \text{softmax}(\mathbf{R}_{fr}^T \times (\mathbf{W}_{td,en} \times \mathbf{h}_i + \mathbf{b}_{td,en})) \\ \mathbf{o}_{en} &= \mathbf{R}_{en} \times \mathbf{p}_{en} \\ \mathbf{o}_{fr} &= \tanh(\mathbf{W}_{t,en} \times (\mathbf{R}_{fr} \times \mathbf{p}_{fr}) + \mathbf{b}_{t,en}) \end{aligned}$$

where $\{\mathbf{W}_{t,en}, \mathbf{b}_{t,en}\}$ are for dimensionality reduction and changing the language space of the query vector $\mathbf{h}_i$ and the context vector, while $\{\mathbf{W}_{td,en}, \mathbf{b}_{td,en}\}$ are only for dimensionality reduction. $\mathbf{o}_{en}$ and $\mathbf{o}_{fr}$ are further combined using a gating mechanism as in Eq. 1 to obtain the final target context vector $\mathbf{o}_{tgt}$ (dimensions *H*).

## 4.3 Dual Conversation History

Now that we have explained how to leverage the source and target conversation history separately, we explain how they can be utilised simultaneously. The simplest way to do this is to incorporate both context vectors $\mathbf{o}_{src}$ and $\mathbf{o}_{tgt}$ into the base model (explained in Sec 4.4), referred as *Src-Tgt* dual context.

Another intuitive approach, as evident from Figure 2, is to separately model English and Foreign sentences using two separate context vectors $\mathbf{o}_{en,m}$ and $\mathbf{o}_{fr,m}$, where each is constructed from a mixture of the original source or target translations, is language-specific and possibly contain less noise. We refer to this as the *Src-Tgt-Mix* dual context. Suppose $\mathbf{R}_{en,m}$ contains the mixed source/target representations for English (the dimensions for source representations have been reduced to $H$) and $\mathbf{R}_{fr,m}$ contains the same for Foreign. Then,

$$\begin{aligned}
\mathbf{p}_{en,m} &= \text{softmax}(\mathbf{R}^T_{en,m} \times (\mathbf{W}_{td,en} \times \mathbf{h}_i + \mathbf{b}_{td,en})) \\
\mathbf{p}_{fr,m} &= \text{softmax}(\mathbf{R}^T_{fr,m} \times \tanh(\mathbf{W}_{tt,en} \times \mathbf{h}_i + \mathbf{b}_{tt,en})) \\
\mathbf{o}_{en,m} &= \tanh(\mathbf{W}_{tr,en} \times (\mathbf{R}_{en,m} \times \mathbf{p}_{en,m}) + \mathbf{b}_{tr,en}) \\
\mathbf{o}_{fr,m} &= \mathbf{R}_{fr,m} \times \mathbf{p}_{fr,m}
\end{aligned}$$

where $\mathbf{W}_{td,en}$, $\mathbf{W}_{tr,en}$ and $\mathbf{W}_{tt,en}$ are for dimensionality reduction, changing the language space and both, respectively.

### 4.4 Incorporating Context into Base Model

The final representations $\mathbf{o}_{src}$ and $\mathbf{o}_{tgt}$ or $\mathbf{o}_{en,m}$ and $\mathbf{o}_{fr,m}$, can be incorporated together or individually in the base model by:

- **InitDec** Using a non-linear transformation to initialise the decoder, similar to Wang et al. (2017): $\mathbf{s}_{i,0} = \tanh(\mathbf{V} \times \mathbf{o}_i + \mathbf{b}_s)$, where $i$ is the sentence index in current turn $2k+1$, $\{\mathbf{V}, \mathbf{b}_s\}$ are encoder-decoder specific parameters and $\mathbf{o}_i$ is either a single context vector or a concatenation (transformed) of the two.

- **AddDec** As an auxiliary input to the decoder (similar to Jean et al. (2017); Wang et al. (2017); Maruf and Haffari (2018)):

  $$\mathbf{s}_{i,n} = \tanh(\boldsymbol{W}_s \cdot \mathbf{s}_{i,n-1} + \boldsymbol{W}_{sn} \cdot \boldsymbol{E}_T[y_{i,n}] + \boldsymbol{W}_{sc} \cdot \boldsymbol{c}_{i,n} + \boldsymbol{W}_{ss} \cdot \mathbf{o}_{i,src} + \boldsymbol{W}_{st} \cdot \mathbf{o}_{i,tgt})$$

- **InitDec+AddDec** Combination of previous two approaches.

### 4.5 Training and Decoding

The model parameters are trained end-to-end by maximising the sum of log-likelihood of the bilingual conversations in training set $\mathcal{D}$. For example, for a conversation having alternating turns of English and Foreign language, the log-likelihood is:

$$\sum_{k=0}^{\frac{|T|}{2}-1} \Big( \sum_{i=1}^{|t_{2k+1}|} \log P_{\boldsymbol{\theta}}(\boldsymbol{y}_i|\boldsymbol{x}_i, \mathbf{o}_i) + \sum_{j=1}^{|t_{2k+2}|} \log P_{\boldsymbol{\theta}}(\boldsymbol{x}_j|\boldsymbol{y}_j, \mathbf{o}_j) \Big)$$

where $i$, $j$ denote sentences belonging to $2k+1^{th}$ or $2k+2^{th}$ turn; $\mathbf{o}_{(.)}$ is a representation of the conversation history, and $|T|$ is the total number of turns (assumed to be even here).

The best output sequence for a given input sequence for the $i^{th}$ sentence at test time, a.k.a. decoding, is produced by:

$$\arg\max_{\boldsymbol{y}_i} P_{\boldsymbol{\theta}}(\boldsymbol{y}_i|\boldsymbol{x}_i, \mathbf{o}_i)$$

## 5 Experiments

**Implementation and Hyperparameters** We implement our conversational Bi-MSMT model in C++ using the DyNet library (Neubig et al., 2017). The base model is built using `mantis` (Cohn et al., 2016) which is an implementation of the generic sentence-level NMT model using DyNet.

The base model has single layer bidirectional GRUs in the encoder and 2-layer GRU in the decoder[8]. The hidden dimensions and word embedding sizes are set to 256, and the alignment dimension (for the attention mechanism in the decoder) is set to 128.

**Models and Training** We do a stage-wise training for the base model, i.e., we first train the English→Foreign architecture and the Foreign→English architecture, using the sentence-level parallel corpus. Both architectures have the same vocabulary[9] but separate parameters to avoid biasing the embeddings towards the architecture trained last. The contextual model is pre-trained similar to training the base model. The best model is chosen based on minimum overall perplexity on the bilingual dev set.

For the source context representations, we use the sentence representations generated by two sentence-level bidirectional RNNLMs (one each for English and Foreign) trained offline. For the target sentence representations, we use the last hidden states of the decoder generated from the pre-trained base model[10]. At decoding time, however, we use the last hidden state of the decoder computed by our model (not the base) as the target sentence representations. Further training details are provided in Appendix B.

[8] We follow Cohn et al. (2016) and Britz et al. (2017) in choosing hyperparameters for our model.

[9] For each language-pair, we use BPE (Sennrich et al., 2016) to obtain a joint vocabulary of size ≈30k.

[10] Even though the paramaters of the base model are updated, the target sentence representations are fixed throughout training. We experimented with a scheduled updating scheme in preliminary experiments but it did not yield significant improvement.

|  | Europarl |  |  |  |  |  |  |  |  | Subtitles |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | En-Fr |  |  | En-Et |  |  | En-De |  |  | En-Ru |  |  |
|  | Overall | En→Fr | Fr→En | Overall | En→Et | Et→En | Overall | En→De | De→En | Overall | En→Ru | Ru→En |
| *Base Model* | 37.36 | 38.13 | 36.03 | 20.68 | 18.64 | 26.65 | 24.74 | 21.80 | 27.74 | 19.05 | 14.90 | 23.04 |
| *+Source Context as Lang-Specific Attention via* |  |  |  |  |  |  |  |  |  |  |  |  |
| InitDec | 38.40$^\dagger$ | 39.19$^\dagger$ | 36.86$^\dagger$ | **21.79**$^\dagger$ | 19.54$^\dagger$ | **28.33**$^\dagger$ | **26.34**$^\dagger$ | **23.31**$^\dagger$ | 29.39$^\dagger$ | 18.88 | 14.89 | 22.56 |
| AddDec | 38.50$^\dagger$ | **39.35**$^\dagger$ | 36.98$^\dagger$ | 21.65$^\dagger$ | **19.66**$^\dagger$ | 27.48$^\dagger$ | 26.30$^\dagger$ | 23.09$^\dagger$ | **29.52**$^\dagger$ | 19.34 | 15.16 | 23.12 |
| InitDec+AddDec | **38.55**$^\dagger$ | 39.34$^\dagger$ | **37.14**$^\dagger$ | 21.49$^\dagger$ | 19.43$^\dagger$ | 27.55$^\dagger$ | 26.25$^\dagger$ | 23.18$^\dagger$ | 29.30$^\dagger$ | **19.35** | **15.16** | **23.14** |
| *+Source Context via* |  |  |  |  |  |  |  |  |  |  |  |  |
| Direct Tranformation | 38.35$^\dagger$ | 39.13$^\dagger$ | 36.96$^\dagger$ | 21.75$^\dagger$ | **19.59**$^\dagger$ | 28.07$^\dagger$ | 26.29$^\dagger$ | 23.34$^\dagger$ | 29.22$^\dagger$ | 19.09 | 14.89 | 22.76 |
| Hierarchical Gating | 38.33$^\dagger$ | 39.14$^\dagger$ | 36.89$^\dagger$ | 21.62$^\dagger$ | 19.55$^\dagger$ | 27.64$^\dagger$ | 26.31$^\dagger$ | 23.17$^\dagger$ | 29.45$^\dagger$ | 19.20 | 15.10 | 22.73 |
| Lang-Specific Attention | 38.40$^\dagger$ | 39.19$^\dagger$ | 36.86$^\dagger$ | 21.79$^\dagger$ | 19.54$^\dagger$ | 28.33$^\dagger$ | 26.34$^\dagger$ | 23.31$^\dagger$ | 29.39$^\dagger$ | **19.35** | **15.16** | **23.14** |
| Combined Attention | **38.50**$^\dagger$ | **39.36**$^\dagger$ | 36.94$^\dagger$ | 21.66$^\dagger$ | 19.52$^\dagger$ | 27.90$^\dagger$ | 26.38$^\dagger$ | 23.31$^\dagger$ | 29.44$^\dagger$ | 18.96 | 14.82 | 22.92 |
| Lang-Specific S-Attention | 38.46$^\dagger$ | 39.24$^\dagger$ | **37.06**$^\dagger$ | **21.84**$^\dagger$ | 19.58$^\dagger$ | **28.43**$^\dagger$ | **26.49**$^\dagger$ | **23.49**$^\dagger$ | **29.49**$^\dagger$ | 19.09 | 14.59 | 22.98 |
| *+Lang-Specific S-Attention using* |  |  |  |  |  |  |  |  |  |  |  |  |
| Source Context | 38.46$^\dagger$ | 39.24$^\dagger$ | 37.06$^\dagger$ | **21.84**$^\dagger$ | 19.58$^\dagger$ | **28.43**$^\dagger$ | **26.49**$^\dagger$ | **23.49**$^\dagger$ | 29.49$^\dagger$ | 19.09 | 14.59 | 22.98 |
| Target Context | 38.76$^\dagger$ | **39.57**$^\dagger$ | 37.35$^\dagger$ | 21.77$^\dagger$ | **19.68**$^\dagger$ | 27.86$^\dagger$ | 26.21$^\dagger$ | 23.16$^\dagger$ | 29.26$^\dagger$ | 19.23 | 14.77 | **23.23** |
| Dual Context Src-Tgt | **38.80**$^\dagger$ | 39.51$^\dagger$ | **37.50**$^\dagger$ | 21.74$^\dagger$ | 19.60$^\dagger$ | 27.98$^\dagger$ | 26.39$^\dagger$ | 23.28$^\dagger$ | **29.50**$^\dagger$ | 18.89 | 14.52 | 23.06 |
| Dual Context Src-Tgt-Mix | 38.76$^\dagger$ | 39.52$^\dagger$ | 37.43$^\dagger$ | 21.68$^\dagger$ | 19.63$^\dagger$ | 27.71$^\dagger$ | 26.37$^\dagger$ | 23.26$^\dagger$ | 29.48$^\dagger$ | **19.26** | **14.86** | 23.01 |

Table 2: BLEU scores for the bilingual test sets. Here all contexts are incorporated as InitDec for Europarl and InitDec+AddDec for Subtitles unless otherwise specified. **bold**: Best performance, †: Statistically significantly better than the base model, based on bootstrap resampling (Clark et al., 2011) with $p < 0.05$.

### 5.1 Results

Firstly, we evaluate the three strategies for incorporating context: InitDec, AddDec, InitDec+AddDec, and report the results for source context using Language-Specific Attention in Table 2. For the Europarl data, we see decent improvements with InitDec for En-Et (+1.11 BLEU) and En-De (+1.60 BLEU), and with InitDec+AddDec for En-Fr (+1.19 BLEU). We also observe that, for all language-pairs, both translation directions benefit from context, showing that our training methodology was indeed effective. On the other hand, for the Subtitles data, we see a maximum improvement of +0.30 BLEU for InitDec+AddDec . We narrow down to three major reasons: (i) the data is noisier when compared to Europarl, (ii) the sentences are short and generic with only 1% having more than 27 tokens, and finally (iii) the turns in OpenSubtitles2016 are short compared to those in Europarl (see Table 1), and we show later (Section 5.2) that the context from current turn is the most important.

The next set of experiments evaluates the five different approaches for computing the source-side context. It is evident from Table 2 that for English-Estonian and English-German, our model indeed benefits from using the fine-grained sentence-level information (Language-Specific Sentence-level Attention) as opposed to just the turn-level one.

Finally, our results with source, target and dual contexts are reported. Interestingly, just using the source context is sufficient for English-Estonian and English-German. For English-French, on the other hand, we see significant improvements for the models using the target-side conversation history over using only the source-side. We attribute this to the base model being more efficient and able to generate better translations for En-Fr as it had been trained on a larger corpus as opposed to the other two language-pairs. Unlike Europarl, for Subtitles, we see improvements for our Src-Tgt-Mix dual context variant over the Src-Tgt one for En→Ru, showing this to be an effective approach when the target representations are noisier.

To summarise, for majority of the cases our Language-Specific Sentence-level Attention is a winner or a close second. Using the Target Context is useful when the base model generates reasonable-quality translations; otherwise, using the Source Context should suffice.

**Local Source Context Model** Most of the previous works for online context-based NMT consider only a single previous sentence as context (Jean et al., 2017; Bawden et al., 2017; Voita et al., 2018). Drawing inspiration from these works, we evaluate our model (trained with Language-Specific Sentence-Level Attention) on the same

|  | Europarl |  |  | Subtitles |
|---|---|---|---|---|
|  | En-Fr | En-Et | En-De | En-Ru |
| *Prev Sent* | 38.15 | 21.70 | 26.09 | **19.13** |
| Our Model | **38.46**† | **21.84** | **26.49**† | 19.09 |

Table 3: BLEU scores for the bilingual test sets. **bold**: Best performance, †: Statistically significantly better than the contextual baseline.

| Type of Context | BLEU |
|---|---|
| No context (Base Model) | 24.74 |
| Current Turn | 26.39 |
| Current Language from Previous Turns | 26.21 |
| Other Language from Previous Turns | 26.32 |
| Complete Context | **26.49** |

Table 4: BLEU scores for En-De bilingual test set.

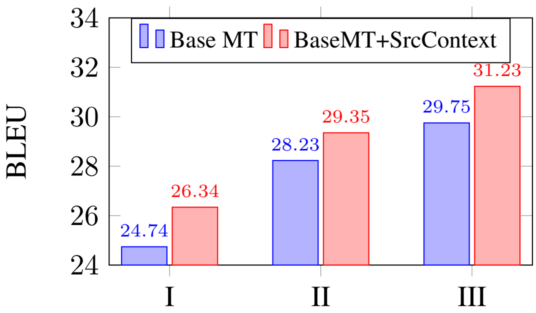


Figure 3: BLEU scores on En-De test set while training (I) smaller base model with smaller corpus (previous experiment), (II) smaller base model with larger corpus, and (III) a larger base model with larger corpus.

test set but using only the previous source sentence as context. This evaluation allows us to hypothesise how much of the gain can be attributed to the previous sentence. From Table 3, it can be seen that our model surpasses the local-context baseline for Europarl showing that the wider context is indeed beneficial if the turn lengths are longer. For En-Ru, it can be seen that using previous sentence is sufficient due to short turns (see Table 1).

### 5.2 Analysis

**Ablation Study** We conduct an ablation study to validate our hypothesis of using the complete context versus using only one of the three types of contexts in a bilingual multi-speaker conversation: (i) current turn, (ii) previous turns in current language, and (iii) previous turns in the other language. The results for En-De are reported in Table 4. We see decrease in BLEU for all types of contexts with significant decrease when considering only current language from previous turns.The results show that the current turn has the most influence on translating a sentence, and we conclude that since our model is able to capture the complete context, it is generalisable to any conversational scenario.

|  |  |
|---|---|
| En→Fr | les; par; est; a; dans; le; en; j'; un; afin; question; entre; qu'; être; ces; également; y; depuis; c'; ou |
| Fr→En | this; of; we; issue; europe; by; up; make; united; does; what; regard; s; must; however; such; whose; share; like; been |
| En→Et | eest; vahel; üle; nimel; ja; aastal; aasta; neid; ainult seepärast; nagu; kes; komisjoni; tehtud; küsimuses; sisserände; liikmesriigi; mulla; liibanoni; dawit |
| Et→En | for; this; of; is; political; important; culture; also; as; order; are; each; their; only; gender; were; its; economy; one; market |
| En→De | daß; auf; und; werden; nicht; müssen; aus; mehr; können; einem; rates; eines; insbesondere; wurden; habe; mitgliedstaaten; ist; sondern; europa; gemeinsamen |
| De→En | that; its; say; must; some; therefore; more; countries; an; favour; public; will; without; particularly; hankiss; much; increase; eu; them; parliamentary |

Table 5: Most frequent tokens correctly generated by our model when compared to the base model.

**Training base model with more data** To analyse if the context is beneficial even when using more data, we perform an experiment for English-German where we train the base model with additional sentence-pairs from the full WMT'14 corpus[11] (excluding our dev/test sets and filtering sentences with more than 100 tokens). For training the contextual model, we still use the bilingual multi-speaker corpus. We observe a significant improvement of +1.12 for the context-based model (Figure 3 II), showing the significance of conversation history in this experiment condition.[12]

We perform another experiment where we use a larger base model, having almost double the number of parameters than our previous base model (hidden units and word embedding sizes set to 512, and alignment dimension set to 256), to test if the model parameters are being overestimated due to the additional context. We use the same WMT'14 corpus to train the base model and achieve significant improvement of +1.48 BLEU for our context-based model over the larger baseline (Figure 3 III).

[11] https://nlp.stanford.edu/projects/nmt/

[12] It should be noted that the BLEU score for the base model trained with WMT does not match the published results exactly as the test set contains both English and German sentences. It does, however, fall between the scores usually obtained on WMT'14 for En→De and De→En.

| | |
|---|---|
| *Context* | nous sommes également favorables au principe d'un système de collecte des miles commun pour le parlement européen, pour que celui-ci puisse bénéficier de billets d'avion moins chers, même si nous voyons difficilement comment ce système pourrait être déployé en pratique.<br>enfin, nous ne sommes pas opposés à l'attribution de prix culturels par le parlement européen. |
| *Source* | néanmoins, nous sommes particulièrement critiques à l'égard du prix pour le journalisme du parlement européen et nous ne pensons pas que celui-ci puisse décerner des prix aux journalistes ayant pour mission de soumettre le parlement européen à un regard critique. |
| *Target* | however, we are highly critical of parliament's prize for journalism, and do not believe that it is appropriate for parliament to award prizes to journalists whose task it is to critically examine the european parliament. |
| *Base Model* | nevertheless, we are particularly critical of the price for the european union's european alism and we do not believe that it would be able to make a price to the journalists who have been made available to the european parliament to a critical view. |
| *Our Model* | however, we are particularly critical of the price for the european union's democratic alism and we do not believe that it can give rise to the prices for journalists who have been tabled to submit the european parliament to a critical view. |

Table 6: Example En-Fr sentence translation showing how the context helps our model in generating the appropriate discourse connective.

| | |
|---|---|
| *Context* | oleks hea, kui reitinguagentuurid vastutaksid tulevikus enda tegevuse eest rohkem.<br>...<br>kirjalikult. - (it) kiites heaks wolf klinzi raporti, mille eesmärk on reitinguagentuuride tõhus reguleerimine, võtab parlament järjekordse sammu finantsturgude suurema läbipaistvuse suunas.<br>...<br>mul oli selle dokumendi üle hea meel, sest krediidireitingute valdkonnal on palju probleeme, millest kõige suuremad on oligopolidele tüpilised struktuurid ning konkurentsi, vastutuse ja läbipaistvuse puudumine. |
| *Source* | selles suhtes tuleb rõhutada nende tegevuse suuremal äbipaistvuse põhirolli. |
| *Target* | in this respect, it is necessary to highlight the central role of increased transparency in their activities. |
| *Base Model* | in this regard it must be emphasised in the major role of transparency in which these activities are to be strengthened. |
| *Our Model* | in this regard, it must be stressed in the key role of greater transparency in their activities. |

Table 7: Example En-Et translation showing how the wide-range context helps in generating the correct pronoun. The antecedent and correct pronoun are highlighted in blue.

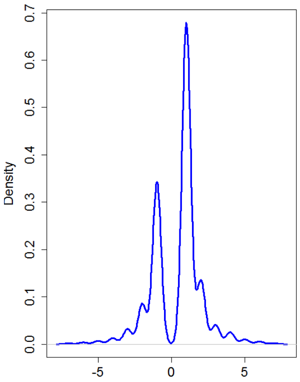


Figure 4: Density of token counts for En→Fr illustrating where our model is better (+ve x-axis) and where the base model is better (-ve x-axis).

**How is the context helping?** The underlying hypothesis for this work is that discourse phenomenon in a conversation may depend on long-range dependency and these may be ignored by the sentence-based NMT models. To analyse if our contextual model is able to accurately translate such linguistic phenomenon, we come up with our own evaluation procedure. We aggregate the tokens correctly generated by our model and those correctly generated by the baseline over the entire test set. We then take the difference of these counts and sort them[13]. Table 5 reports the top 20 tokens where our model is better than the baseline for the Europarl dataset. Figure 4 gives the density of counts obtained using our evaluation for En→Fr[14]. Positive counts correspond to correct translations by our model while the negative counts correspond to where the base model was better. It can be seen that for majority of cases our model supersedes the base model. We observed a similar trend for other translation directions. In general, the correctly generated tokens by our model include pronouns (that, this, its, their, them), discourse connectives (e.g., 'however', 'therefore', 'also') and prepositions (of, for, by).

Table 6 reports an example where our model is able to generate the correct discourse connective '*however*' using the context. If we look at the con-

[13] We do not normalise the counts with the background frequency as it favours rare words. Thus, obscuring the main reasons of improving the BLEU score.

[14] Outliers and tokens with equal counts for our model and the baseline were removed.

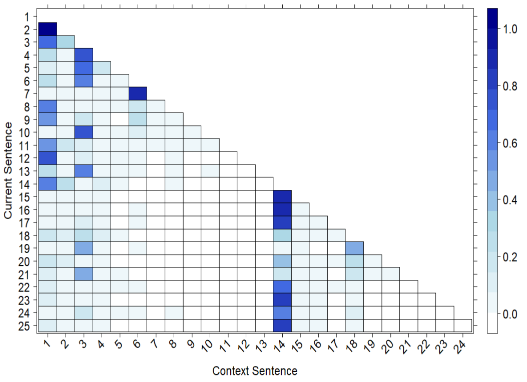


Figure 5: Attention map when translating a conversation from the Et-En test set.

text of the source sentence in French, we come to the conclusion that 'however' is indeed a perfect fit in this case, whereas the base model is at a disadvantage and completely changes the underlying meaning of the sentence by generating the inappropriate connective 'nevertheless'.

Table 7 gives an instance where our model is able to generate the correct pronoun '*their*'. It should be noted that in this case, the current source sentence does not contain the antecedent and thus the context-free baseline is unable to generate the appropriate pronoun. On the other hand, our contextual model is able to do so by giving the highest attention weights to sentences containing the antecedent (observed from the attention map in Figure 5)[15]. Figure 5 also shows that for translating majority of the sentences, the model attends to wide-range context rather than just the previous sentence, hence strengthening the premise of using the complete context.

## 6 Conclusion

This work investigates the challenges associated with translating multilingual multi-speaker conversations by exploring a simpler task referred to as Bilingual Multi-Speaker Conversation MT. We process Europarl v7 and OpenSubtitles2016 to obtain an introductory dataset for this task. Compared to models developed for similar tasks, our work is different in two aspects: (i) the history captured by our model contains multiple languages, and (ii) our model captures 'global' history as opposed to 'local' history captured in most previous works. Our experiments demonstrate the significance of leveraging the bilingual conversation history in such scenarios. Furthermore, the analysis shows that using wide-range context, our model generates appropriate pronouns and discourse connectives in some cases. We hope this work to be a first step towards translating multilingual multi-speaker conversations. Future work on this task may include optimising the base translation model and approaches that condition on specific discourse information in the conversation history.

[15] For this particular conversation, all previous turns were in Estonian.

## Acknowledgments

We thank the reviewers for their useful feedback that helped us in improving our work. We also thank Pierre Lison for providing us with the speaker-annotated OpenSubtitles2016 dataset. This work is supported by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) (www.massive.org.au), the European Research Council (ERC StG DeepSPIN 758969), the Fundação para a Ciência e Tecnologia through contract UID/EEA/50008/2013, a Google Faculty Research Award to GH and the Australian Research Council through DP160102686.

## References

Parnia Bahar, Tamer Alkhouli, Jan-Thorsten Peter, Christopher Jan-Steffen Brix, and Hermann Ney. 2017. Empirical investigation of optimization algorithms in neural machine translation. In *Conference of the European Association for Machine Translation*, pages 13–26, Prague, Czech Republic.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2017. Evaluating discourse phenomena in neural machine translation. In *Proceedings of NAACL–HLT 2018*.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Kyunghyun Cho, B van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. 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)*, pages 176–181. Association for Computational Linguistics.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885. Association for Computational Linguistics.

Eva Martínez Garcia, Carles Creus, Cristina España-Bonet, and Lluís Màrquez. 2017. Using word embeddings to enforce document-level lexical consistency in machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108:85–96.

Eva Martínez Garcia, Cristina España-Bonet, and Lluís Màrquez. 2014. Document-level machine translation as a re-translation process. *Procesamiento del Lenguaje Natural*, 53:103–110.

Zhengxian Gong, Min Zhang, and Guodong Zhou. 2011. Cache-based document-level statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 909–919. Association for Computational Linguistics.

Christian Hardmeier and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In *International Workshop on Spoken Language Translation*, pages 283–289.

Christian Hardmeier, Joakim Nivre, and Jörg Tiedemann. 2012. Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190. Association for Computational Linguistics.

Cong Duy Vu Hoang, Trevor Cohn, and Gholamreza Haffari. 2016. Incorporating side information into recurrent neural network language models. In *Proceedings of NAACL–HLT 2016*, pages 1250–1255.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? In *arXiv:1704.05135*.

Gang Ji and Jeff Bilmes. 2004. Multi-speaker language modeling. In *Proceedings of HLT–NAACL 2004*.

Yangfeng Ji, Trevor Cohn, Lingpeng Kong, Chris Dyer, and Jacob Eisenstein. 2015. Document context language models. In *Workshop track - ICLR 2016*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Conference Proceedings: the 10th Machine Translation Summit*, pages 79–86. AAMT.

Shaohui Kuang, Deyi Xiong, Weihua Luo, and Guodong Zhou. 2018. Modeling coherence for neural machine translation with dynamic and topic caches. *COLING 2018*.

Pierre Lison and Raveesh Meena. 2016. Automatic turn segmentation of movie & tv subtitles. In *Proceedings of the 2016 Spoken Language Technology Workshop*, pages 245–252, San Diego, CA, USA. IEEE.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from Movie and TV subtitles. In *Proceedings of the $10^{th}$ International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the $54^{th}$ Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Quan Hung Tran, Ingrid Zuckerman, and Gholamreza Haffari. 2016. Inter-document contextual language model. In *Proceedings of NAACL–HLT 2016*, pages 762–766.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2017. Learning to remember translation history with a continuous cache.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2816–2821. Association for Computational Linguistics.

Longyue Wang, Xiaojun Zhang, Zhaopeng Tu, Andy Way, and Qun Liu. 2016. Automatic construction of discourse corpora for dialogue translation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2016. Measuring the effect of conversational aspects on machine translation quality. In *Proceedings of COLING 2016*, pages 2571–2581.

## A Data Statistics

| | **Europarl** | | | **Subtitles** |
|---|---|---|---|---|
| | **En-Fr** | **En-Et** | **En-De** | **En-Ru** |
| | | **Dev/Test** | | |
| # Conversations | 140/209 | 88/132 | 70/108 | 462/694 |
| # Sentences | 4.9k/7.8k | 3.2k/5.2k | 2.1k/3.3k | 5.9k/9k |

Table 8: General statistics for development and test sets.

## B Experiments

**Training** For the base model, we make use of stochastic gradient descent (SGD)[16] with initial learning rate of 0.1 and a decay factor of 0.5 after the fifth epoch for a total of 15 epochs. For the contextual model, we use SGD with an initial learning rate of 0.08 and a decay factor of 0.9 after the first epoch for a total of 30 epochs. To avoid overfitting, we employ dropout and set its rate to 0.2. To reduce the training time of our contextual model, we perform computation of one turn at a time, for instance, when using the source context, we run the Turn-RNNs for previous turns once and re-run the Turn-RNN only for sentences in the current turn.

[16] In our preliminary experiments, we tried SGD, Adam and Adagrad as optimisers, and found SGD to achieve better perplexities in lesser number of epochs (Bahar et al., 2017).

# Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation

**Antonio Toral**
Center for Language and Cognition
University of Groningen
The Netherlands
a.toral.ruiz@rug.nl

**Sheila Castilho** **Ke Hu** **Andy Way**
ADAPT Centre
Dublin City University
Ireland
firstname.secondname@adaptcentre.ie

## Abstract

We reassess a recent study (Hassan et al., 2018) that claimed that machine translation (MT) has reached human parity for the translation of news from Chinese into English, using pairwise ranking and considering three variables that were not taken into account in that previous study: the language in which the source side of the test set was originally written, the translation proficiency of the evaluators, and the provision of inter-sentential context. If we consider only original source text (i.e. not translated from another language, or translationese), then we find evidence showing that human parity has not been achieved. We compare the judgments of professional translators against those of non-experts and discover that those of the experts result in higher inter-annotator agreement and better discrimination between human and machine translations. In addition, we analyse the human translations of the test set and identify important translation issues. Finally, based on these findings, we provide a set of recommendations for future human evaluations of MT.

## 1 Introduction

Neural machine translation (NMT) has revolutionised the field of MT by overcoming many of the weaknesses of the previous state-of-the-art phrase-based machine translation (PBSMT) (Bentivogli et al., 2016; Toral and Sánchez-Cartagena, 2017). In only a few years since the first working models, this approach has led to a substantial improvement in translation quality, reported in terms of automatic metrics (Bojar et al., 2016, 2017; Sennrich et al., 2016). This has ignited higher levels of expectation, fuelled in part by hyperbolic claims from large MT developers. First we saw in Wu et al. (2016) that Google NMT was "bridging the gap between human and machine translation [quality]". This was amplified recently by the claim by Hassan et al. (2018) that Microsoft had ”achieved human parity" in terms of translation quality on news translation from Chinese to English, and more recently still by SDL who claimed to have "cracked" Russian-to-English NMT with "near perfect" translation quality.[1] However, when human evaluation is used to compare NMT and SMT, the results do not always favour NMT (Castilho et al., 2017a,b).

Accompanying the claims regarding the capability of the Microsoft Chinese-to-English NMT system, Hassan et al. (2018) released their experimental data[2] which permits replicability of their experiments. In this paper, we provide a detailed examination of Microsoft's claim to have reached *human parity* for the task of translating news from Chinese (ZH) to English (EN). They provide two definitions in this regard, namely:

**Definition 1**. *If a bilingual human judges the quality of a candidate translation produced by a human to be equivalent to one produced by a machine, then the machine has achieved human parity.*

**Definition 2**. *If there is no statistically significant difference between human quality scores for a test set of candidate translations from a machine translation system and the scores for the corresponding human translations then the machine has achieved human parity.*

The remainder of the paper is organised as follows. First, we identify and discuss three potential issues in Microsoft's human evaluation, concerning (i) the language in which the source text was originally written, (ii) the competence of the human evaluators with respect to translation, and (iii) the linguistic context available to these evaluators (Section 2). We then conduct a new modified

[1] https://www.sdl.com/about/news-media/press/2018/sdl-cracks-russian-to-english-neural-machine-translation.html

[2] http://aka.ms/Translator-HumanParityData

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 113–123
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64012

evaluation of their MT system on the same dataset taking these issues onboard (Section 3). In so doing, we reassess whether human parity has indeed been achieved following what we consider to be a fairer evaluation setting. We then take a closer look at the quality of Microsoft's dataset with the help of an English native speaker and a Chinese native speaker, and discover a number of problems in this regard (Section 4). Finally, we conclude the paper (Section 5) with a set of recommendations for future human evaluations, together with some remarks on the risks for the whole field of overhyping the capability of the systems we build.

## 2 Potential Issues

### 2.1 Original Language of the Source Text

The test set used by Hassan et al. (2018) (`newstest2017`) was the ZH reference from the news translation shared task at WMT 2017 (Bojar et al., 2017),[3] which contains 2,001 sentence pairs, of which half were originally written in ZH and the remaining half were originally written in EN. Figure 1 represents the WMT test set and the respective translation. The organisers of WMT 2017 manually translated each of these two subsets (files A1 and B1 in Figure 1) into the other language (B2 and A2, respectively) to produce the resulting parallel test set of 2,001 sentence pairs. Thus, Hassan et al. (2018) machine-translated 2,001 sentences from ZH into EN, but only half of them were originally written in ZH (file D1); the other half were originally written in EN, then they were translated by a human translator into ZH (as part of WMT's organisation), and this human translation was finally machine-translated by Microsoft into EN (file D2). Microsoft also human-translated the ZH reference file into EN to use as reference translations (file C - EN REF). Therefore, 50% of their EN reference comprises EN translations direct from the original Chinese (file C1), while 50% are EN translations from the human-translated file from EN into ZH (file C2), i.e. backtranslation of the original EN (A1). While their human evaluation is conducted on three different subsets (referred to as Subset-2, Subset-3, and Subset-4 in Tables 5d to 5f of their paper), since all three are randomly sampled from the whole test set, these subsets still contain around 50% of sentences originally written in ZH and around 50% originally written in EN.

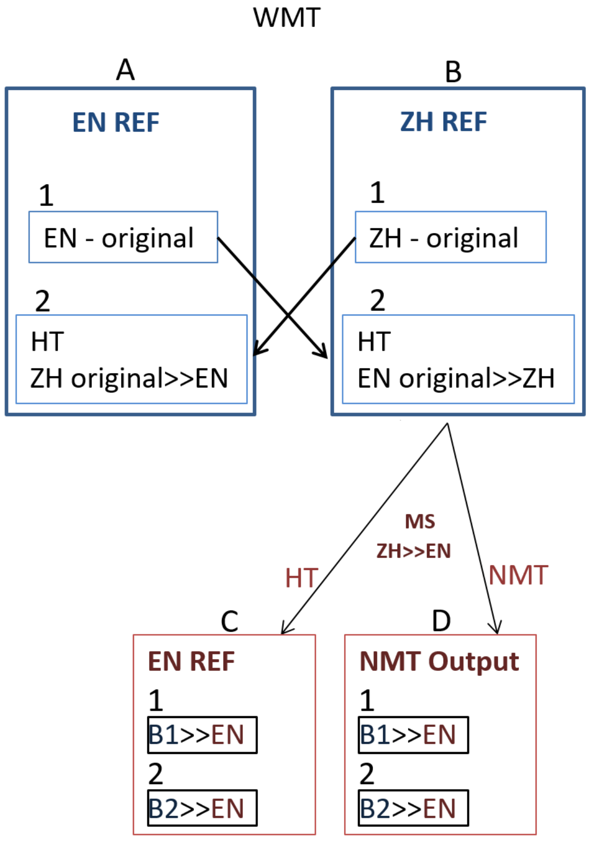


Figure 1: WMT test set and Microsoft Translation ZH-to-EN reference and MT output

We hypothesize that the sentences originally written in EN are easier to translate than those originally written in ZH, due to the simplification principle of translationese, namely that translated sentences tend to be simpler than their original counterparts (Laviosa-Braithwaite, 1998). Two additional universal principles of translation, explicitation and normalisation, would also indicate that a ZH text originally written in EN would be easier to translate. Therefore, we explore whether the inclusion of source ZH sentences originally written in EN distorts the results, and unfairly favours MT.

### 2.2 Human Evaluators

The human evaluation described in Hassan et al. (2018) was conducted by "bilingual crowd workers". While the authors implemented a set of quality controls to "guarantee high quality results", no further details are provided on the selection of evaluators and their linguistic expertise. In addition, no inter-annotator agreement (IAA) figures were provided. We acknowledge, however, that agreement cannot be measured using the conven-

[3] http://www.statmt.org/wmt17/translation-task.html

tional Kappa coefficient, since their human evaluation uses a continuous scale (range $[0-100]$).

It has been argued that non-expert translators lack knowledge of translation and so might not notice subtle differences that make one translation better than another. This was observed in the human evaluation of the TraMOOC project[4] in which authors compared the evaluation of MT output of professional translators against crowd workers (Castilho et al., 2017c). Results showed that for all language pairs (involving 11 languages), the crowd workers tend to be more accepting of the MT output by giving higher fluency and adequacy scores and performing very little post-editing.

With that in mind, we attempt to replicate the results achieved in Hassan et al. (2018) by redoing the manual evaluation with participants with different levels of translation proficiency, namely professional translators (henceforth referred to as experts) and bilingual speakers with no formal translation qualifications (henceforth referred to as non-experts).

### 2.3 Context

Hassan et al. (2018) evaluated the sentences in the testset in randomised order, meaning that sentences were evaluated in isolation. However, documents such as the news stories that make up the test set contain relations that go beyond the sentence level. To translate them correctly one needs to take this inter-sentential context into account (Voigt and Jurafsky, 2012; Wang et al., 2017a). The MT system by Hassan et al. (2018) translates sentences in isolation while humans naturally consider the wider context when conducting translation.

Our hypothesis is that referential relations that go beyond the sentence-level were ignored in the evaluation as its setup considered sentences in isolation (randomised). This probably resulted in the evaluation missing some errors by the MT system that might have been caused by its lack of inter-sentential contextual knowledge. In contrast, our revised human evaluation takes inter-sentential context into account. Sentences are not randomised but evaluated in the order they appear in the documents that make up the test set. In addition, when a sentence is evaluated, the evaluator can see both the previous and the next sentence, akin to how a professional translator works in practice. In the same spirit, concurrent work by Läubli et al. (2018) contrasts the evaluation of single sentences and entire documents in the dataset by Hassan et al. (2018), and shows a stronger preference for human translation over MT when evaluating documents as compared to isolated sentences.

## 3 Evaluation

### 3.1 Experimental Setup

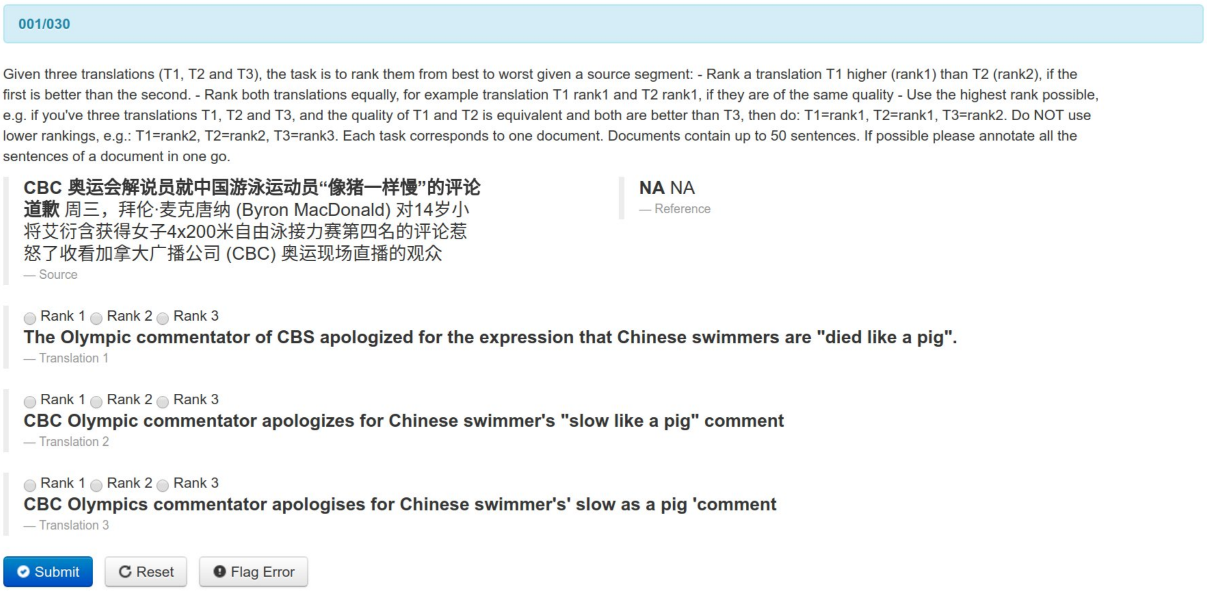
We conduct a human evaluation in which at the same time evaluators are shown a source ZH sentence and three EN translations thereof: (i) the human translation produced by Microsoft (file C in Figure 1: henceforth referred to as HT), (ii) the output of Microsoft's MT system (file D: henceforth MS), and the output of a production system, Google Translate (henceforth GG).[5] We take these three translations from the data provided by Hassan et al. (2018).

Instead of giving evaluators randomly selected sentences, they see them in order. We randomised the documents in the test set (169) and prepared one evaluation task per document, for the first 49 documents (503 sentences). Of these 49 documents, 41 were originally written in ZH (amounting to 299 sentences, with each document containing 7.3 sentences on average) and the remaining 8 were originally written in EN (204 sentences, average of 25.5 sentences per document). Evaluators were asked to annotate all the sentences of each document in one go, so that they can take inter-sentential context into account.

Rather than direct assessment (DA) (Graham et al., 2015), as in Hassan et al. (2018), we conduct a relative ranking evaluation. While DA has some advantages over ranking and has replaced the latter at the WMT shared task since 2017 (Bojar et al., 2017), ranking is more appropriate for our evaluation due to the fact that we evaluate sentences in consecutive order (rather than randomly). This can be accommodated in ranking as we can show all three translations for each source sentence together with the previous and next source sentences

[4] http://tramooc.eu/

[5] We note that in the study by Hassan et al. (2018), 9 different translations were compared: 3 reference translations, and the output from six MT systems, 4 of which were Microsoft systems (including one online), plus Google Translate and the Sogou system (Wang et al., 2017b), the best-performing system at WMT-2017. This, together with the fact that we use different methods, may affect the comparability of the results obtained to some degree.



001/030

Given three translations (T1, T2 and T3), the task is to rank them from best to worst given a source segment: - Rank a translation T1 higher (rank1) than T2 (rank2), if the first is better than the second. - Rank both translations equally, for example translation T1 rank1 and T2 rank1, if they are of the same quality - Use the highest rank possible, e.g. if you've three translations T1, T2 and T3, and the quality of T1 and T2 is equivalent and both are better than T3, then do: T1=rank1, T2=rank1, T3=rank2. Do NOT use lower rankings, e.g.: T1=rank2, T2=rank2, T3=rank3. Each task corresponds to one document. Documents contain up to 50 sentences. If possible please annotate all the sentences of a document in one go.

**CBC 奥运会解说员就中国游泳运动员“像猪一样慢”的评论道歉** 周三，拜伦·麦克唐纳 (Byron MacDonald) 对14岁小将艾衍含获得女子4x200米自由泳接力赛第四名的评论惹怒了收看加拿大广播公司 (CBC) 奥运现场直播的观众
— Source

**NA** NA
— Reference

○ Rank 1 ○ Rank 2 ○ Rank 3
**The Olympic commentator of CBS apologized for the expression that Chinese swimmers are "died like a pig".**
— Translation 1

○ Rank 1 ○ Rank 2 ○ Rank 3
**CBC Olympic commentator apologizes for Chinese swimmer's "slow like a pig" comment**
— Translation 2

○ Rank 1 ○ Rank 2 ○ Rank 3
**CBC Olympics commentator apologises for Chinese swimmer's' slow as a pig 'comment**
— Translation 3

Figure 2: Snapshot from the human evaluation showing the first sentence from the first document, which contains 30 sentences.

at the same time. In contrast, in DA only one translation is shown at a time, which is of course evaluated in isolation. An important advantage of DA is that the number of annotations required grows linearly (rather than exponentially with ranking) with the number of translations to be evaluated; this is relevant for WMT's shared task as there may be many MT systems to be evaluated, but not for our research as we have only three translations (HT, MS and GG). In any case, both approaches have been found to lead to very similar outcomes as their results correlate very strongly ($R \geq 0.92$ in Bojar et al. (2016)).

Our human evaluation is performed with the Appraise tool (Federmann, 2012).[6] Figure 2 shows a snapshot of the evaluation. Subsequently, we derive an overall score for each translation (HT, MS and GG) based on the rankings. To this end we use the TrueSkill method adapted to MT evaluation (Sakaguchi et al., 2014) following its usage at WMT15,[7] i.e. we run 1,000 iterations of the rankings recorded with Appraise followed by clustering (significance level $\alpha = 0.05$).

Five evaluators took part in our evaluation: two professional Chinese-to-English translators and three non-experts. Of the two professional translators, one is a native English speaker with a fluent level of Chinese, and the other is a Chinese native speaker with a fluent level of English. The three non-expert bilingual participants are Chinese native speakers with an advanced level of English. These bilingual participants are researchers in NLP, and so their profile is similar to some of the human evaluators of WMT, namely MT researchers.[8]

All evaluators completed all 49 documents, except the third non-expert, who completed the first 18. Similarly, all evaluators ranked all the sentences in the documents they evaluated, except the second professional translator, who skipped 3 sentences. In total we collected 6,675 pairwise judgements.

## 3.2 Results

### 3.2.1 Original Language

To find out whether the language in which the source sentence was originally written has any effect on the evaluation, we show the resulting Trueskill scores for each translation taking into account all the sentences in our test set versus considering the sentences in two groups according to the original language (ZH and EN). The results are shown in Table 1.

Regardless of the original language, GG is the lowest-ranked translation, thus providing an indi-

[6] https://github.com/cfedermann/Appraise

[7] https://github.com/mjpost/wmt15

[8] It is an open question as to whether using bilingual NLP researchers may affect the results obtained. While we follow the practice of WMT here – which differs from the approach taken by Hassan et al. (2018), who used bilingual crowd workers – we intend in future work to investigate this further.

| Rank | Original language | | |
|---|---|---|---|
| | Both<br>$n = 6675$ | ZH<br>$n = 3873$ | EN<br>$n = 2802$ |
| 1 | HT 1.587* | HT 1.939* | MS 1.059 |
| 2 | MS 1.231* | MS 1.199* | HT 0.772* |
| 3 | GG -2.819 | GG -3.144 | GG -1.832 |

Table 1: Ranks of the translations given the original language of the source side of the test set shown with their Trueskill score (the higher the better). An asterisk next to a translation indicates that this translation is significantly better than the one in the next rank.

cation that the quality obtainable from the MS system is a notable improvement over state-of-the-art NMT systems used in production. We observe that HT outperforms significantly MS when the original language is ZH, but the difference between the two is not significant when the original language is EN. Hence, we confirm our hypothesis that the use of translationese as the source language distorts the results in favour of MS.

Next, we check whether this effect of translationese is also present in the evaluation by Hassan et al. (2018). To this end, we concatenate all their judgments and model them with mixed-effects regression. Our dependent variable is the score, scaled down from the original range $[0, 100]$ to $[0, 1]$, which we aim to predict with one continuous predictor – sentence length – and two factorial independent variables: translation (levels HT, MS and GG) and original language (levels ZH and EN). The identifiers of the sentence and the annotator are included as random effects. We plot the interaction between the translation and the original language of the resulting model in Figure 3. HT outperforms MS by around 0.05 absolute points for sentences whose original language is ZH. However this gap disappears for source sentences originally written in EN, where we see that the score for MS is actually slightly higher than that of HT, though the difference is not significant. We observe a clear effect of translationese (EN): compared to ZH, the scores of both MT systems increase substantially (GG over 10% absolute and MS over 6% absolute), while the HT score increases only very slightly.

Our hypothesis was theoretically supported by the simplification principle of translationese. Applied to the test data, this would mean that the portion originally written in ZH is more complex than the part originally written in EN. To check whether this is the case, we compare the two subsets of the test set using a measure of text complexity, type-token ratio (TTR). While both subsets contain a similar number of sentences (1,001 and 1,000), the ZH subset contains more tokens (26,468) than its EN counterpart (22,279). We thus take a subset of the ZH (840 sentences) containing a similar amount of words to the EN data (22,271 words). We then calculate the TTR for these two subsets using bootstrap resampling. The TTR for ZH ($M = 0.1927$, $SD = 0.0026$, 95% confidence interval $[0.1925, 0.1928]$) is 13% higher than that for EN ($M = 0.1710$, $SD = 0.0025$, 95% confidence interval $[0.1708, 0.1711]$).

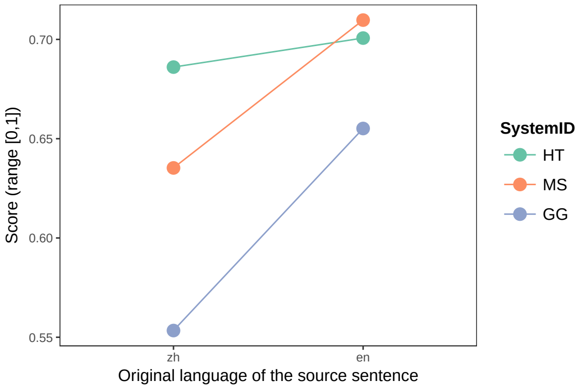


Figure 3: Interaction between the MT system (levels HT, MS and GG) and the original language of the source sentence (levels ZH and EN).

Given the findings of this experiment, in the remainder of the paper we use only the subset of the test set whose original language is ZH.

### 3.2.2 Evaluators

To find out whether the translation expertise of the evaluator has any effect on the evaluation, we show the resulting Trueskill scores for each translation resulting from the evaluations by non-expert versus expert translators. The results are shown in Table 2. The gap between HT and MS is considerably wider for experts (2.2 vs 1.2) than for non-experts (1.3 vs 0.9). We link this to our expectation, based on the previous finding by Castilho et al. (2017c), that non-experts are more lenient regarding MT errors. In other words, non-experts disregard translation subtleties in their evaluation, which leads to the gap between different translations – in this case HT and MS – being smaller. In Section 4 we explore this further by means of a qualitative analysis.

| Rank | Translators | | |
|---|---|---|---|
| | **All**<br>$n = 3873$ | **Experts**<br>$n = 1785$ | **Non-experts**<br>$n = 2088$ |
| 1 | HT 1.939* | HT 2.247* | HT 1.324 |
| 2 | MS 1.199* | MS 1.197* | MS 0.94* |
| 3 | GG -3.144 | GG -3.461 | GG -2.268 |

Table 2: Ranks and Trueskill scores (the higher the better) of the three translations for evaluations carried out by expert versus non-expert translators. An asterisk next to a translation indicates that this translation is significantly better than the one in the next rank.

Trueskill provides not only an overall score for each translation but also its confidence interval. We expect these to be wider for the annotations by non-experts than those annotations given by experts, which would indicate that there is more uncertainty in the rankings by non-experts. Figure 4 shows the scores for each translation by experts and non-experts, i.e. the same values that were shown in Table 2, now enriched with their 95% confidence intervals.

The sum of the confidence scores for the three translations is just 0.33% higher for non-experts (3.076) than for experts (3.066). However, it is worth mentioning that, compared to the width of the intervals for experts, those for non-experts are considerably wider for HT (16% relative difference) while they are similar or smaller for MT (1% and -11% relative differences for GG and MS, respectively).

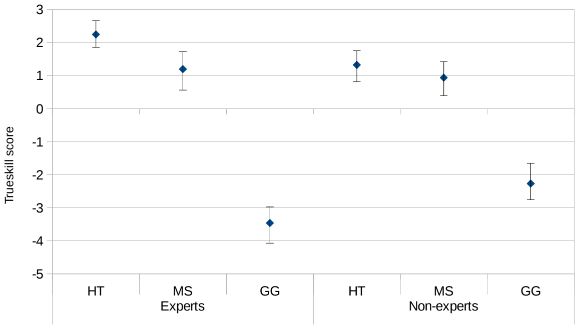


Figure 4: Trueskill scores of the three translations by experts and non-experts together with their confidence intervals.

We now look at inter-annotator agreement (IAA) between experts and non-experts. We compute the Kappa ($\kappa$) coefficient (Cohen, 1960), as done at WMT 2016 (Bojar et al., 2016, Section 3.3):[9]

$$k = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(A)$ represents the proportion of times that the annotators agree, and $P(E)$ the proportion of times that the annotators are expected to agree by chance.

As expected, the IAA between professional translators ($\kappa = 0.254$) is notably higher, 95% relative, than that between non-experts ($\kappa = 0.130$).[10] As we have three non-experts, we can calculate the IAA not only among the three of them but also between all three pairs of non-expert annotators; all of the resulting coefficients (0.057, 0.135 and 0.195) are lower than that between experts (0.254).

To the best of our knowledge, this is the first time that IAA of professional translators and non-experts has been compared for the human evaluation of MT. In related work, Callison-Burch (2009) compared the agreement level of two types of non-expert translators: MT developers (referred to in that paper as 'experts') and crowd workers. He showed that crowd workers can reach the agreement level of MT researchers using multiple workers and weighting their judments. That said, both types of non-experts conducted human evaluations for WMT13 (Bojar et al., 2013) and the IAA rates of the crowd were well below those of the researchers.

## 4 Analyses

As mentioned previously, we have examined the quality of the test sets, both originally written in ZH and originally written in EN and their respective translations. An English native speaker analysed both the original EN version from the WMT set (file A1 in Figure 1) and the human translation of the set originally written in ZH performed by Microsoft (file C2). A Chinese native speaker, who is fluent in English and has experience with translation from EN into ZH, analysed the original

[9] https://github.com/cfedermann/wmt16/blob/master/scripts/compute_agreement_scores.py

[10] Due to the fact that one non-expert evaluated only 18 out of the 49 documents, the IAA calculations consider only the first 18 documents. If we consider all 49 documents, the trend remains the same; the IAA for the two experts is higher than that for the two non-experts who evaluated all the documents: 0.265 vs 0.196.

ZH versions (file B1) as well as the human translation of the set originally written in EN performed by the WMT organisers (file B2).

### 4.1 Original English

Regarding the English original (file A1 in Figure 1), the analysis showed that apart from a few grammar errors, the test set appeared to be fluent and grammatical. Examples of grammatical errors in the original EN files are:

**i)** “The idiot didn’t realize they were still on the air”
**ii)** “Soon after, Scott Russel who was hosting CBC’s broadcast apologized on-air for MacDonald’s comment, saying: ‘We apologize the comment on a swim performance made it to air.’ ”

In example i) “on air” should be used instead of “on the air”, while in the example ii) a missing “that” should be used after “apologize”. Nonetheless, these errors did not affect the ZH translation (file B2) or the following backtranslation (C2) into EN. Our hypothesis is that because the test set is news content, it also contains tweets (such as example i)) and quotes from speech interviews (such as example ii)), which are more likely to contain grammatical errors.

### 4.2 Chinese Translation

Regarding the human translation into ZH performed by WMT (file B2 in Figure 1), most of the sentences contained grammatical errors and/or mistranslations of proper nouns. Furthermore, although some translations were grammatically correct and accurate, they were not fluent. When the ZH-translated sentences were compared against the source (A1), the translations were mostly accurate. However, when analyzed on their own without the source, they sound disfluent:

**iii)**
EN original (A1): A front-row seat to the stunning architecture of the Los Angeles Central Library
ZH (B2):洛杉矶中央图书馆的惊艳结构先睹为快

**iv)**
EN original (A1): An open, industrial loft in DTLA gets a cozy makeover
ZH (B2): DTLA的开放式工厂阁楼进行了一次舒适的改造。

In example iii), although the ZH translation has fully transferred the meaning of the source text, it contains word-order errors which makes the translation disfluent since the verb phrase “先睹为快” (take a look) is placed after the object (library). One possible translation for that is “抢先目睹洛杉矶中央图书馆的惊艳结构” because the ZH language syntax requires the verb to be placed before the object.

In example iv), the ZH translation contains a grammatical error in the word “进行”, which would imply that the loft is carrying out a makeover. In addition, the adjective “舒适的” (cosy) cannot be used to describe “改造” (makeover). One possible translation for the English sentence is “DTLA的开放式工业阁楼被改造的很舒适”.

Given this analysis, we speculate that the translation of the EN original files into ZH might not have been performed by an experienced translator, but rather exemplify either human translation performed by an inexperienced translator, or poorly post-edited MT.

### 4.3 English Translation

Regarding the EN reference files translated by Microsoft (file C2 in Figure 1), many of the sentences contained grammatical errors (such as word order, verb tense and missing prepositions) as well as mistranslations.

**v)**
EN original (A1): A front-row seat to the stunning architecture of the Los Angeles Central Library
ZH (B2):洛杉矶中央图书馆的惊艳结构先睹为快
EN (C2): Take a look of the astounding architecture of the Los Angeles Central Library.

GG: The stunning structure of the Los Angeles Central Library
MS: A sneak peek at the stunning architecture of the Los Angeles Central Library

**vi)**
EN original (A1): An open, industrial loft in DTLA gets a cozy makeover
ZH (B2):DTLA的开放式工厂阁楼进行了一次舒适的改造。
EN (C2): A comfortable makeover was provided

to the open factory building design of DTLA.

GG: DTLA's Open factory loft has a comfortable makeover.
MS: DTLA's open-plan factory loft has undergone a comfortable makeover.

In example v), the EN translation of the ZH source[11] analyzed previously is translated with the wrong preposition, i.e. 'look of' instead of 'look at'. None of the professional translators considered the reference worse than the MS output; while one translator and one non-expert considered it 'as good' as the MS output, the other considered it better than MS but worse than GG. Regarding the non-expert assessment, two of them considered the HT to be as good as MS and better than GG, and one considered the HT to be worse than MS but better than GG.

In example vi), the EN translation (C2) of the ZH source (B2) does not have all the information expressed in ZH as the word 'loft' (阁楼) is not translated. Moreover, the EN translation refers to an architectural design makeover of the building rather than an interior makeover of an attic. Both professional translators considered the EN reference to be worse than the MS output. As far as the non-experts are concerned, two of them considered the HT to be worse than MS, while one considered it to be 'as good'. This provides qualitative evidence that non-experts may be more tolerant of translation errors than professional translators.

Another example of such behaviour is the following:

**vii)**
EN original (A1): Learn more about the history of downtown's Central Library as the Society of Architectural Historians/Southern California Chapter hosts a salon with Arnold Schwartzman and Stephen Gee, authors of the new book "Los Angeles Central Library: A History of its Art and Architecture
ZH (B2): 美国建筑史学家学会南加利福尼亚洲分会与新书《洛杉矶中央博物馆：其艺术与建筑历史》的作者阿诺·斯瓦茨曼和史蒂芬·吉举办了一场沙龙。观众们可通过此次活动进一步了解市中心中央图书馆的历史

EN (C2):A salon will be hosted by Southern California Branch of Society of Architectural Historians and the co-authors of Los Angles Central Museum: Art and Architectural History, Arnold Schwarzman and Stephen Gee. It will deliver more knowledge of the Central Library to the participants

GG: The Southern California branch of the American Institute of Architectural Historians has held a salon with 阿诺·斯瓦茨曼 and 史蒂芬·吉, author of the Los Angeles Central Museum: its art and architectural history. Through this event, viewers can learn more about the history of Central Library in the city centre
MS: The Southern California chapter of the American Society of Architectural Historians and the authors of a new book, "Los Angeles Central Museum: Its Art and Architectural History," Arnold Schwartzman and Steven Gee, hosted a salon at which viewers learned more about the history of the Central Library in the city center

In example vii), regarding the ZH source (B2), in addition to having the first sentence translated into past tense – whereas the EN original (A1) shows the salon event is happening in the near future – it also contains a typo '洲' which means 'continent' instead of 'state' '州'. Even though the typo does not affect the EN translation (C2), it shows that the quality of the ZH translation is not as high as would be expected of professional human translators. Regarding the EN translation (C2), while the first sentence is mostly fluent – although it contains a typo in 'Angles' (Angeles) and lacks the article 'the' before the proper noun in the first sentence – the second sentence lacks fluency and contains errors of omissions and mistranslations. For example, the words "downtown" and "history" (市中心 and 历史, respectively) were not transferred over to the EN reference (C2). Furthermore, the word 'viewers' in the ZH translation (观众们) was mistranslated as 'participants'. Nonetheless, the EN translation (C2) is able to capture the correct tense of the sentence since the second sentence in the ZH translation (B2) is ambiguous regarding verbal tense. The MS translation does a better job in keeping the fluency throughout the sentence even though it mis-

[11] It is important to note that the translators did not have access to the original EN (A1) and so the ZH file (B2) was used as the source.

translates the tense of the source in the past tense. Both professional translators assessed the HT as worse than MS, whereas two of the non-experts considered it to be as good as MS and better than GG. The third non-expert considered the HT to be worse than both MT systems. This example shows that the level of expertise of the evaluators may have an effect on the evaluation given that non-experts are wrongly more tolerant of MT errors.

Similarly to the ZH translation (B2) of the English original, we speculate that the EN translation (C2) of the ZH files is more likely a human translation performed by an inexperienced translator, or even a poorly post-edited machine translation; even if the translation was performed by an experienced translator, such that the ZH source (B2) contained errors or was disfluent, a professional translator would surely be more meticulous and fix such errors before rubber-stamping the translations.

## 5 Conclusions and Future Work

This paper has reassessed a recent study that claimed that MT has reached human parity for the translation of news from Chinese into English, considering three variables that were not taken into account in that previous study: (i) the language in which the source side of the test set was originally written, (ii) the translation proficiency of the evaluators, and (iii) the provision of inter-sentential context.

The main findings of this paper are the following:

- If we consider the subset of the test set whose source side was originally written in ZH, there is evidence that human parity has not been achieved, i.e. the difference between the human and the machine translations is significant. This is the case both in our human evaluation and in Microsoft's.

- Having translationese (ZH translated from EN in our study) as input, compared to having original text, results in higher scores for MT systems in Microsoft's human evaluation.

- Compared to judgments by non-experts, those by professional translators have a higher IAA and a wider gap between human and machine translations.

- We have identified issues in the human translations by both WMT and Microsoft. These indicate that these translations were conducted by non-experts and that were possibly post-edited MT output.

There is little doubt that human evaluation has played a very important role in MT research and development to date. As MT systems improve – as exemplified by the progress made by Hassan et al. (2018) over state-of-the-art production systems – and thus the gap between them and human translators narrows, we believe that human evaluation, in order to remain useful, needs to be more discriminative. We suggest that a set of principles should be adhered to, partly based on our findings, which we outline here as recommendations:

- The original language in which the source side of the test sets is written should be the same as their source language. This will avoid having spurious effects because of having translationese as MT input.

- Human evaluations should be conducted by professional translators. This allows fine-grained nuances of translations to be taken into account in the evaluation and should result in higher inter-annotator agreement.

- Human evaluations should proceed taking the whole document into account rather than evaluating sentences in isolation. This allows for intersentential phenomena to be considered as part of the evaluation.

- Test sets should be translated by experienced professional translators from scratch.

We are confident that adhering to these principles is sensible under any translation conditions. Of course, if the test set is faulty, then in claiming that one's MT system outperforms one's competitors, there is a risk that what one is actually demonstrating is the contrary, as if automatic evaluation metrics demonstrate a higher score, what that could be denoting is that one's output is actually closer to the faulty test set than producing better output in terms of improved translation quality *per se*. Of course, this has consequences not just for the study in this paper, but for all shared tasks: past, present, and future.[12]

[12] Ideally, it would be great if multiple references were also

Should material be made available by Google, SDL or any other MT developers who claim 'human parity' or the like, we would be very happy to apply these principles in subsequent rigorous evaluations of actual demonstrable improvements in translation quality. One thing is certain; as Way (2018) observes, "those of us who have seen many paradigms come and go know that overgilding the lily does none of us any good, especially those of us who have been trying to build bridges between MT developers and the translation community for many years." We trust that our findings in this paper demonstrate that while MT quality does seem to be improving quite dramatically, human translators will continue to find gainful employment for many years to come, despite somewhat grandiose claims to the contrary.

On a final note, we acknowledge that our conclusions and recommendations are somewhat limited in that they are derived from experiments on just one language direction and five evaluators. Therefore we plan as future work to conduct similar experiments on additional language pairs with a higher number of evaluators. In the spirit of Hassan et al. (2018), without which this paper would not have been possible, we too make publicly available our evaluation materials, the anonymised human judgments and the statistical analyses thereof.[13]

## Acknowledgments

We would like to thank the five expert and non-expert translators that took part in this study. We are also grateful for valuable comments on this paper from Hany Hassan, the lead author on the Microsoft paper. This research was partially supported by the iADAATPA project funded by CEF research action (2016-EU-IA-0132) under grant agreement No.1331703. The ADAPT Centre for Digital Content Technology at Dublin City University is funded under the Science Foundation Ireland Research Centres Programme (Grant 13/RC/ 2106) and is co-funded under the European Regional Development Fund.

## References

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, Texas.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany.

Chris Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 286–295, Singapore.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley, and Andy Way. 2017a. Is neural machine translation the new state of the art? *The Prague Bulletin of Mathematical Linguistics*, 108(1):109–120.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Rico Sennrich, Vilelmini Sosoni, Panayota Georgakopoulou, Pintu Lohar, Andy Way, Antonio Valerio Miceli Barone, and Maria Gialama. 2017b. A Comparative Quality Evaluation of PBSMT and NMT using Professional Translators. In *MT Summit 2017*, pages 116–131, Nagoya, Japan.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Andy Way, Panayota Georgakopoulou, Maria Gialama, Vilelmini Sosoni, and Rico Sennrich. 2017c. Crowdsourcing for NMT evaluation: Professional

available, but the point remains that if these are poor quality human translations, then this is likely to skew results still further.

[13] https://github.com/antot/human_parity_mt

translators versus the crowd. In *Translating and the Computer 39*, London. https://www.asling.org/tc39/?page_id=3223.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Christian Federmann. 2012. Appraise: An open-source toolkit for manual evaluation of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Yvette Graham, Timothy Baldwin, and Nitika Mathur. 2015. Accurate evaluation of segment-level machine translation metrics. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1183–1191, Denver, Colorado.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, Will Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving Human Parity on Automatic Chinese to English News Translation. https://arxiv.org/abs/1803.05567.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has Machine Translation Achieved Human Parity? A Case for Document-level Evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

Sara Laviosa-Braithwaite. 1998. Universals of translation. In *Routledge Encyclopedia of Translation Studies*, pages 288–291. Routledge, London.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient Elicitation of Annotations for Human Evaluation of Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 1–11, Baltimore, Maryland, USA.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation (WMT16)*, pages 371–376, Berlin, Germany.

Antonio Toral and Víctor M. Sánchez-Cartagena. 2017. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1063–1073, Valencia, Spain.

Rob Voigt and Dan Jurafsky. 2012. Towards a literary machine translation: The role of referential cohesion. In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 18–25, Montrèal, Canada.

Longyue Wang, Zhaopeng Tu, Andy Way, and Liu Qun. 2017a. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017b. Sogou neural machine translation systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, WMT 2017*, page 410–415, Copenhagen, Denmark.

Andy Way. 2018. Machine translation: Where are we at today? In Angelone E, Massey G, and Ehrensberger-Dow M, editors, *The Bloomsbury Companion to Language Industry Studies*. Bloomsbury, London. In press.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. https://arxiv.org/abs/1609.08144.

## Appendix: Evaluator Instructions

Given three translations (T1, T2 and T3), the task is to rank them from best to worst given a source segment:

- Rank a translation T1 higher (rank1) than T2 (rank2), if the first is better than the second.
- Rank both translations equally, for example translation T1 rank1 and T2 rank1, if they are of the same quality.
- Use the highest rank possible, e.g. if you've three translations T1, T2 and T3, and the quality of T1 and T2 is equivalent and both are better than T3, then do: T1=rank1, T2=rank1, T3=rank2. Do NOT use lower rankings, e.g.: T1=rank2, T2=rank2, T3=rank3.

Each task corresponds to one document. Documents contain up to 50 sentences. If possible please annotate all the sentences of a document in one go.

# Freezing Subnetworks to Analyze Domain Adaptation in Neural Machine Translation

**Brian Thompson**[†] **Huda Khayrallah**[†] **Antonios Anastasopoulos**[‡]
**Arya D. McCarthy**[†] **Kevin Duh**[†] **Rebecca Marvin**[†] **Paul McNamee**[†]
**Jeremy Gwinnup**[°] **Tim Anderson**[°] and **Philipp Koehn**[†]
[†]Johns Hopkins University, [‡]University of Notre Dame, [°]Air Force Research Laboratory
{brian.thompson, huda, arya, becky, mcnamee, phi}@jhu.edu,
aanastas@nd.edu, kevinduh@cs.jhu.edu,
{jeremy.gwinnup.1, timothy.anderson.20}@us.af.mil

## Abstract

To better understand the effectiveness of continued training, we analyze the major components of a neural machine translation system (the encoder, decoder, and each embedding space) and consider each component's contribution to, and capacity for, domain adaptation. We find that freezing any single component during continued training has minimal impact on performance, and that performance is surprisingly good when a single component is adapted while holding the rest of the model fixed. We also find that continued training does not move the model very far from the out-of-domain model, compared to a sensitivity analysis metric, suggesting that the out-of-domain model can provide a good generic initialization for the new domain.

## 1 Introduction

Neural Machine Translation (NMT) has supplanted Phrase-Based Machine Translation (PBMT) as the standard for high-resource machine translation. This has necessitated new domain adaptation methods, because PBMT adaptation methods primarily rely on adapting the language model and phrase table using interpolation or back-off schemes (see §2). Continued training (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016), also referred to as fine-tuning, is one of the most popular methods for NMT adaptation, due to its strong performance.

In contrast to the PBMT literature, little research has focused on why continued training is effective or on what happens to NMT models during continued training. Motivated by domain adaptation analysis in PBMT (Haddow and Koehn, 2012; Duh et al., 2010; Irvine et al., 2013), this work proposes a simple *freezing subnetworks* technique and uses it to gain insight into how the various components of an NMT system behave during continued training.

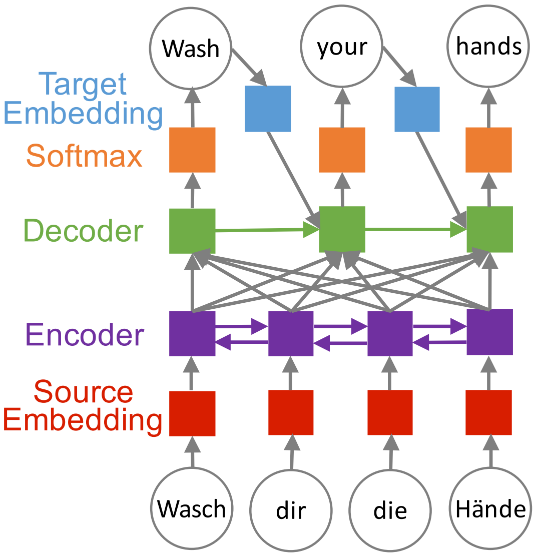


Figure 1: Visualization of an NMT system segmented into components.

We segment the model into five subnetworks, which we refer to as *components*, denoted in Figure 1: the source embeddings, encoder, decoder (which includes the attention mechanism), the softmax (used to denote the decoder output embeddings and biases), and the target embeddings.

We freeze components one at a time during continued training to see how much the adaptation depends on each component. We also experiment with freezing everything except one component to determine each component's capacity to adapt to the new domain on its own.

In order to further analyze continued training, we examine the magnitude of change in model components during continued training of the network, under both normal and freezing training conditions. We also conduct sensitivity analysis of each component to assist in interpreting these magnitudes.

Our NMT adaptation experiments are performed across three languages: we translate from German,

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 124–132
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64013

| Component | Size |
|---|---|
| Target Embedding | 15.1M |
| Softmax | 15.1M |
| Decoder | 6.8M |
| Encoder | 3.7M |
| Source Embedding | 15.4M |
| Total | 56.0M |

Table 1: Number of parameters in each component.

Korean, and Russian into English. Our out-of-domain models are trained on WMT and/or subtitles corpora, and we adapt each model to translate patent abstracts.

## 2 Related Work

Continued training has recently become a standard for domain or cross-lingual adaptation in several neural NLP applications. In PBMT, the most prominent methods focus on adapting the language model component (Moore and Lewis, 2010), and/or the translation model (Matsoukas et al., 2009; Mansour and Ney, 2014; Axelrod et al., 2011), or on interpolating in-domain and out-of-domain models (Lu et al., 2007; Foster et al., 2010; Koehn and Schroeder, 2007).

In contrast, the methods employed in NMT tend to utilize continued training, which involves initializing the model with pre-trained weights (trained on out-of-domain data) and training/adapting it to the in-domain data. Among others, Luong and Manning (2015) and Freitag and Al-Onaizan (2016) applied this method for domain adaptation. Chu et al. (2017) mix in-domain and out-of-domain data during continued training in order to adapt to multiple domains. Continued training has also been applied to cross-lingual transfer learning for NMT, with Zoph et al. (2016) and Nguyen and Chiang (2017) using it for transfer between high- and low-resource language pairs.

Continued training is effective on a range of data sizes. In-domain gains have been shown with as few as dozens of in-domain training sentences (Miceli Barone et al., 2017), and recent work has explored continued training on single sentences (Farajian et al., 2017; Kothur et al., 2018).

Similar adaptation techniques are also employed in the field of Automatic Speech Recognition, where continued training has been the basis of cross-lingual transfer learning approaches (Grézl et al., 2014; Kunze et al., 2017). Usually, the lower layers of the network, which perform acoustic modeling, are frozen and only the upper layers are updated. In a similar vein, other works (Swietojanski and Renals, 2014; Vilar, 2018) adapt a network to a new domain by learning additional weights that re-scale the hidden units.

| | | Tokens | |
|---|---|---|---|
| Dataset | Sentences | Source | Target |
| Out-of-domain training sets | | | |
| Ru–En WMT | 25.2 M | 563.9 M | 595.9 M |
| Ru–En Subtitles | 25.9 M | 179.8 M | 212.4 M |
| De–En WMT | 5.8 M | 138.6 M | 131.8 M |
| De–En Subtitles | 22.5 M | 171.6 M | 185.8 M |
| Ko–En Subtitles | 1.4 M | 11.5 M | 11.9 M |
| In-domain training sets | | | |
| Ru–En WIPO | 29 k | 620 k | 812 k |
| De–En WIPO | 821 k | 19 M | 23 M |
| Ko–En WIPO | 81 k | 2.2 M | 2.0 M |
| In-domain test sets | | | |
| Ru–En WIPO | 3 k | 82 k | 109 k |
| De–En WIPO | 3 k | 132 k | 162 k |
| Ko–En WIPO | 3 k | 187 k | 165 k |

Table 2: Dataset statistics. The number of tokens is computed before segmentation into subwords. The in-domain development sets (not shown) have similar statistics to the test sets.

## 3 Data

Our experiments are carried out across three language pairs, from Russian, Korean, and German into English. Basic statistics on the datasets used for our experiments are summarized in Table 2. The three languages represent three different domain adaptation scenarios:

- In German, both the in- and out-of-domain datasets are large.
- In Russian, the in-domain dataset is large but the out-of-domain dataset is small.
- In Korean, both in- and out-of-domain datasets are small.

| | |
|---|---|
| OpenSubtitles | You're gonna need a bigger boat. |
| WMT | Intensified communication and sharing of information between the project partners enables the transfer of expertise in rural tourism. |
| WIPO | The films coated therewith, in particular polycarbonate films coated therewith, have improved properties with regard to scratch resistance, solvent resistance, and reduced oiling effect, said films thus being especially suitable for use in producing plastic parts in film insert molding methods. |

Table 3: Example sentences to illustrate domain differences.

### 3.1 Out-of-domain Data

For our out-of-domain dataset we utilize the `OpenSubtitles2018` corpus (Tiedemann, 2016; Lison and Tiedemann, 2016), which consists of translated movie subtitles.[1] For De–En and Ru–En, we also use data from WMT 2017 (Bojar et al., 2017),[2] which contains data from several sources: Europarl (parliamentary proceedings) (Koehn, 2005),[3] News Commentary (political and economic news commentary),[4] Common Crawl (web-crawled parallel corpus), and the EU Press Releases.

We use the final 2500 lines of `OpenSubtitles2018` for the development set. For German and Russian we also concatenate `newstest2016` as part of the development set. `newstest2016` consists of translated news articles released by WMT for its shared task. In Korean, we rely only on the `OpenSubtitles2018` data. See Table 3 for example sentences from WMT and OpenSubtitles.

### 3.2 In-domain Data

We perform adaptation into the World International Property Organization (WIPO) COPPA-V2 dataset (Junczys-Dowmunt et al., 2016).[5] The WIPO data consist of parallel sentences from international patent application abstracts. We reserve 3000 lines each for the in-domain development and test sets. See Table 3 for an example WIPO sentence.

### 3.3 Data Preprocessing

All our datasets were tokenized using the Moses[6] tokenizer. Additionally, Korean text was segmented into words using the KoNLPy wrapper of the Mecab-Ko segmenter.[7]

As a final preprocessing step, we train Byte Pair Encoding (BPE) segmentation models (Sennrich et al., 2016) on the out-of-domain training corpus. We train separate BPE models for each language, each with a vocabulary size of 30,000. For each language, BPE is trained on the out-of-domain corpus only and then applied to the training, development, and test data for both out-of-domain and in-domain datasets. This mimics the realistic setting where a generic, computationally-expensive-to-train NMT model is trained once. This NMT model is then adapted to new domains as they emerge, without retraining on the out-of-domain corpus. Training BPE on the in-domain data would change the vocabulary and thus require re-building the model.

## 4 Experimental Setup

For all language pairs, we train systems on the out-of-domain data and select the best model parameters based on perplexity on the out-of-domain development set. We then adapt the systems into our smaller, in-domain training sets. We select the best model based on the WIPO development set perplexity and report results on the WIPO test sets.

### 4.1 Continued Training

We define continued training as:

1. Train a model until convergence on large out-of-domain bitext.
2. Initialize a new model with the final parameters of Step 1.
3. Train the model from Step 2 until convergence on in-domain bitext.

[1] `www.opensubtitles.org`
[2] `statmt.org/wmt17`
[3] `statmt.org/europarl`
[4] `casmacat.eu/corpus/news-commentary.html`
[5] `wipo.int/patentscope/en/data`
[6] `statmt.org/moses/`
[7] `konlpy.org/en/`

### 4.2 NMT Implementation and Settings

Our neural machine translation systems are trained using SOCKEYE (Hieber et al., 2017).[8] We use SOCKEYE's built-in functionality for freezing parameters. We build RNN-based encoder–decoder models with attention (Bahdanau et al., 2015), using a bidirectional RNN for the encoder. The encoder and decoder both have 2 layers with LSTM hidden sizes of 512. Source and target word vectors are also of size 512. The number of parameters in each component are given in Table 1.

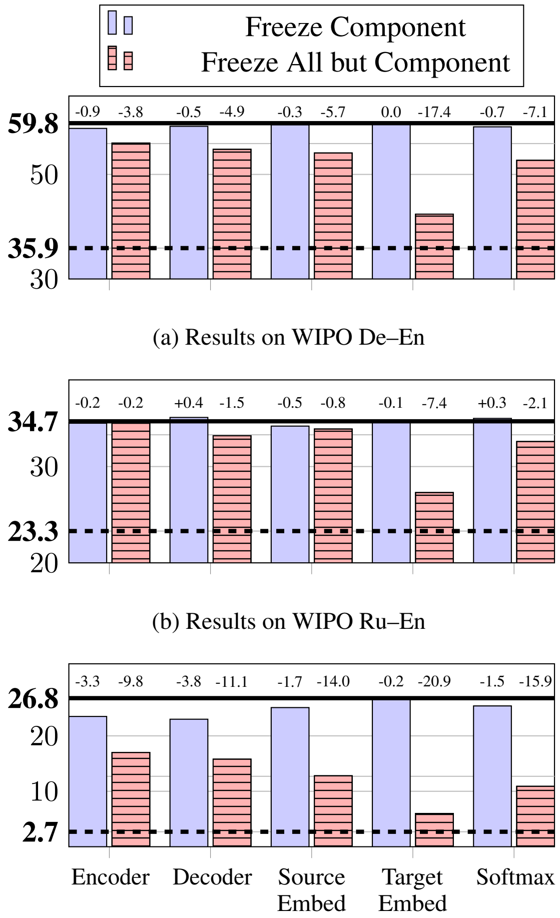
While training the out-of-domain models, we apply dropout with 10% probability on the RNN layers. We apply label smoothing of 0.1. We use ADAM (Kingma and Ba, 2014) as the optimizer, using a learning rate of 0.0003 and a learning rate reduce factor of 0.7. We use a batch size of 4096 words and create a checkpoint every 4000 minibatches.

We do not use dropout or label smoothing during continued training because we do not want regularization to bias our measurements of magnitude changes during continued training (see §5.3). We note, however, that each would likely increase in-domain performance. Our batch size during continued training is 128 sentences, and we create a checkpoint every half epoch. Our learning rate reduce factor for continued training is 0.5. We run each continued training experiment over a set of learning rates (0.1, 0.01, 0.001, 0.0001, 0.00001) and choose the best result based on the perplexity on the development set, as previous work has suggested that even when using ADAM, continued training can be sensitive to learning rate (Farajian et al., 2017; Li et al., 2018; Kothur et al., 2018). We use dot product attention (Luong et al., 2015), which means we do not have a separate attention component; the attention is implicitly built into the decoder.

## 5 Results and Analysis

### 5.1 Freezing One Component at a Time

Our first set of experiments measure the extent to which performance depends on updating any given component in the model. We perform continued training while freezing a single component (i.e. keeping that component fixed to the values from the out-of-domain model used to initialize training while adapting the rest of the components). The results for this setting are shown in the solid left bars of Figure 2.



(a) Results on WIPO De–En

(b) Results on WIPO Ru–En

(c) Results on WIPO Ko–En

Figure 2: BLEU scores when freezing only the denoted component (left solid bars) and when freezing all but the denoted component (right striped bars). The horizontal lines denote baselines: no adaptation (dashed) and full continued training (solid). The labels on top of each bar denote the difference from the full continued training baseline.

For De–En and Ru–En, the out-of-domain models have reasonable performance on the in-domain test set. In these language pairs, freezing any single component has little impact on in-domain BLEU. The worst change is $-0.9$ BLEU—when freezing the De–En encoder—and in some cases we see small gains of up to 0.4 BLEU. We interpret these gains as trivial (and possibly the result of variance) but there may be an NMT continued training scenario in which freezing could increase performance by acting as a regularizer (see Ghahremani et al., 2017).

In Ko–En, where the out-of-domain model does poorly on the in-domain test set, we see more sub-

[8] github.com/awslabs/sockeye

stantial drops when freezing a component during continued training. Freezing the decoder and encoder does the most harm (−3.8 and −3.3 BLEU, respectively), followed by the source embeddings and softmax components (−1.7 and −1.5 BLEU, respectively).

In all cases, freezing the target embeddings has very little impact (at most −0.2 BLEU, in Ko–En), suggesting that it is relatively unimportant during adaptation. These results show that the model and training procedure are very robust; continued training is able to find a local minimum for the new domain which has (nearly) equal performance to the one found in full training, even though an entire component is fixed to the initial out-of-domain model's values.

This robustness suggests that caution is in order when attempting to interpret changes of any single component—in particular, changes in the surrounding components must also be considered. For example, it appears that when the source embeddings are fixed, the encoder is able to compensate for the non-adapted source embeddings and adapt the system to interpret source tokens correctly in the new domain. Conversely, it appears that when the encoder is fixed, the source embeddings are able to adapt to produce vectors for source tokens which are interpreted correctly by the un-adapted encoder. Note that adaptation to source tokens in the new domain could theoretically occur in any un-frozen component, an idea further explored in the next section.

### 5.2 Freezing All But One Component

In our second set of experiments, we freeze all but one component during continued training to see how much each component, in isolation, is able to adapt the NMT system to the new domain. The results are shown in Figure 2 (right striped bars).

We find that only adapting a single component is—somewhat surprisingly—not catastrophic in most cases. Adapting only the encoder, for example, still gives a gain of 20.1 BLEU over the out-of-domain model (3.8 BLEU worse than full continued training) in German and 11.4 BLEU (0.2 BLEU worse than full continued training) in Russian.

In De–En and Ko–En, we see that adapting just the encoder does the best, followed by the decoder, source embeddings, softmax, and target embeddings. The trend in Russian is similar but with the decoder and source embeddings switched.

| | Russian | German | Korean |
|---|---|---|---|
| Softmax | 0.0347 | 0.0578 | 0.0650 |
| Encoder | 0.0236 | 0.0520 | 0.0654 |
| Decoder | 0.0209 | 0.0465 | 0.0594 |
| Source Embed | 0.0165 | 0.0417 | 0.0414 |
| Target Embed | 0.0141 | 0.0357 | 0.0422 |

Table 4: Euclidean distance moved by each component when components are adapted jointly.

| | Russian | German | Korean |
|---|---|---|---|
| Softmax | 0.0345 | 0.2215 | 0.1031 |
| Encoder | 0.0516 | 0.2857 | 0.1494 |
| Decoder | 0.0419 | 0.2751 | 0.1122 |
| Source Embed | 0.0563 | 0.3045 | 0.0893 |
| Target Embed | 0.0714 | 0.2940 | 0.5777 |

Table 5: Euclidean distance moved by each component when components are adapted individually.

These experiments suggest the encoder is most able to adapt the model to a new domain in isolation. It is worth noting that the encoder achieves this despite being the component with the fewest parameters (3.7M). The target embeddings are least able to adapt the model to a new domain (consistent with §5.1).

These experiments also show that the upper bound for adapting a single component is quite high, suggesting that the upper bound for adaptation techniques using monolingual data to adapt individual components could be quite high as well. Of course, it seems unlikely that techniques using only monolingual data can achieve the same level of performance as when directly optimizing on bitext.

### 5.3 Magnitude of Changes During Continued Training

We are interested in the overall magnitude of the changes experienced by each component during continued training, (i.e., how far each moves from the out-of-domain model) and how those changes compare to the cases where only a single component was adapted.

We had two opposing hypotheses that could predict adaptation behavior when only one component is being adapted (as in §5.1):

1. The portion of the network producing the component's input is fixed, as is the portion of the network that interprets the component's output. This suggests the component will be somewhat constrained, in contrast to full continued training where the components may adapt jointly over time.
2. Since all other components are fixed, the adapting component has to bear all the responsibility for changing the entire model's behavior, requiring more drastic changes than it would have undergone during full continued training.

The Euclidean distance between each component in the initial out-of-domain model and the continued training model are shown in Table 4 (normal continued training) and Table 5 (trained individually).[9] While further work would be required to make any definitive statements, the results clearly favor the second hypothesis. The movement of individually adapted components tends to be larger than that of their counterparts in fully adapted models.

### 5.4 Sensitivity Analysis

To assist in interpreting the overall magnitude of changes experienced during continued training, we perform sensitivity analysis of each component of the initial, out-of-domain model. In each experiment, zero-mean, independent Gaussian noise with fixed variance is added to every parameter in a single component of the model. By varying noise levels, we show how much (random) movement is required to produce a given decrease in performance.[10]

Figure 3 shows the sensitivity plots for each component. Table 6 shows, for each component, the (linearly interpolated) BLEU score decrease that would result from adding random noise of the same magnitude as the change observed in full continued training.

| | Russian | German | Korean |
|---|---|---|---|
| Softmax | $-1.29$ | $-3.00$ | $-5.49$ |
| Encoder | $-0.05$ | $-0.78$ | $-1.68$ |
| Decoder | $-0.23$ | $-0.52$ | $-1.05$ |
| Source Embed | $-0.12$ | $-0.10$ | $-0.22$ |
| Target Embed | $-0.08$ | $-0.02$ | $-0.04$ |

Table 6: Sensitivity Analysis: Change in BLEU for random perturbation of magnitude corresponding to the distance each component moved during standard continued training.

Considering the sensitivity of each component reveals several patterns. First, the most significant change in the network, compared to the sensitivity metric, is in the softmax component for all three languages. Second, these values are rather small compared to the overall improvements seen in continued training (+23.0 in De–En, +24.2 in Ko–En, and +11.4 in Ru–En). This suggests that the in-domain model parameters are, on average, fairly close to the out-of-domain model used to initialize training; even though the out-of-domain model does not have a particularly high BLEU score, it is close to a good local minimum in the in-domain error surface.

## 6 Conclusions

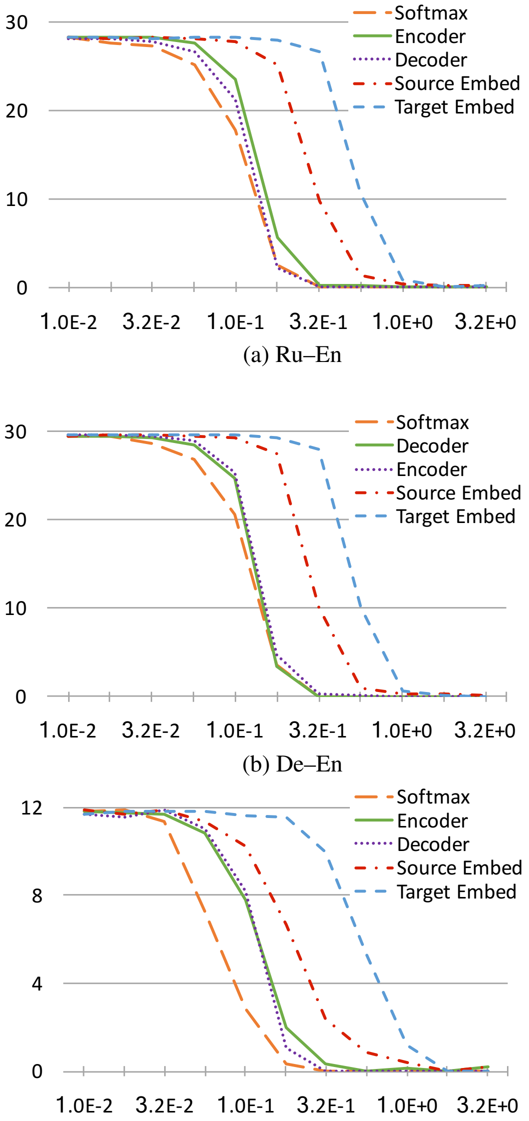
This work presents and applies a simple *freezing subnetworks* method to analyze continued training.

Freezing any single component during continued training has negligible effect on performance compared to full continued training. Furthermore, adapting only a *single* component via continued training produces surprisingly strong performance in most cases, achieving most of the performance gain of full continued training. That is, continued training is able to adapt the overall system to a new domain by modifying only parameters in a single component. This finding goes against the intuitive hypothesis that source embeddings must account for domain changes in the source vocabulary, target embeddings must account for changes in the target vocabulary, etc.

We note that the encoder and decoder, despite having the least parameters (3.7M and 6.8M, respectively, out of 56M), perform strongly across all languages. This suggests further work on adapting only a subset of parameters may be warranted (see also Vilar, 2018; Michel and Neubig, 2018).

[9] To compute this distance, all weights and biases in a given component are concatenated into a vector (i.e. we compute the Frobenius norm).

[10] Bojar et al. (2010) show that very low BLEU scores are not trustworthy. Due to the very low BLEU score (2.7) of the out-of-domain Ko–En system on the in-domain test set, we use out-of-domain test sets for each language, where BLEU scores fall between 11 and 30. This means that the BLEU scores for continued training (computed on the in-domain test set) are not directly comparable to the BLEU scores produced for sensitivity analysis. However, as the sensitivity analysis is used only as an aid in interpreting the general magnitude of BLEU shifts, we view this as an acceptable compromise.



(c) Ko–En

Figure 3: Performance degradation (BLEU) as a function of noise (standard deviation) added to a given component.

We also perform sensitivity analysis of components and find that continued training does not move the model very far from the initial out-of-domain model, in the sense that random perturbations of the same magnitude cause only small performance drops on the out-of-domain test set. This suggests that the out-of-domain model, while not performing very well on the in-domain test set, is close to a good local minimum on the in-domain error surface. This finding may explain the recent success of techniques which regularize a continued training model using the initial, out-of-domain model (Miceli Barone et al., 2017; Dakwale and Monz, 2017; Khayrallah et al., 2018).

## Acknowledgements

The authors would like to thank Lane Schwartz and Graham Neubig for their roles in organizing the MT Marathon in the Americas (MTMA), where this work began. The authors would also like to thank Michael Denkowski and David Vilar for assistance with SOCKEYE. This material is based upon work supported in part by the DARPA LORELEI and IARPA MATERIAL programs. Brian Thompson is supported by the Department of Defense through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. Antonios Anastasopoulos is supported by the National Science Foundation (NSF) Award 1464553.

## References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing*, pages 355–362. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. Proceedings of the International Conference on Learning Representations (ICLR).

Ondřej Bojar, Kamil Kos, and David Mareček. 2010. Tackling sparse data issue in machine translation evaluation. In *Proc. ACL*, pages 86–91. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391. Association for Computational Linguistics.

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 5 Sep 2018. Originator reference number RH-18-118777. Case number 88ABW-2018-4431.

Praveen Dakwale and Christof Monz. 2017. Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data. *Proceedings of the XVI Machine Translation Summit*, page 117.

Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Analysis of translation model adaptation in statistical machine translation. In *International Workshop on Spoken Language Translation (IWSLT) 2010*.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark. Association for Computational Linguistics.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 451–459. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast Domain Adaptation for Neural Machine Translation. *CoRR*, abs/1612.06897.

Pegah Ghahremani, Vimal Manohar, Hossein Hadian, Daniel Povey, and Sanjeev Khudanpur. 2017. Investigation of transfer learning for asr using lf-mmi trained neural networks. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 279–286.

Frantisek Grézl, Martin Karafiát, and Karel Vesely. 2014. Adaptation of multilingual stacked bottleneck neural network structure for new language. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 7654–7658. IEEE.

Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on smt systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1712.05690*.

Ann Irvine, John Morgan, Marine Carpuat, Hal Daumé III, and Dragos Munteanu. 2013. Measuring machine translation errors in new domains. *Transactions of the Association for Computational Linguistics*, 1:429–440.

Marcin Junczys-Dowmunt, Bruno Pouliquen, and Christophe Mazenc. 2016. Coppa v2. 0: Corpus of parallel patent applications building large parallel corpora with gnu make. In *4th Workshop on Challenges in the Management of Large Corpora Workshop Programme*.

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 36–44. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the second workshop on statistical machine translation*, pages 224–227. Association for Computational Linguistics.

Sachith Sri Ram Kothur, Rebecca Knowles, and Philipp Koehn. 2018. Document-level adaptation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 64–73, Melbourne, Australia. Association for Computational Linguistics.

Julius Kunze, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober. 2017. Transfer learning for speech recognition on a budget. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 168–177.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One Sentence One Model for Neural Machine Translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Yajuan Lu, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford Neural Machine Translation Systems for Spoken Language Domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Saab Mansour and Hermann Ney. 2014. Translation model based weighting for phrase extraction. In *Conference of the European Association for Machine Translation*, pages 35–43.

Spyros Matsoukas, Antti-Veikko I Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 708–717. Association for Computational Linguistics.

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494, Copenhagen, Denmark. Association for Computational Linguistics.

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. *arXiv preprint arXiv:1805.01817*.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proc. IJCNLP*, volume 2, pages 296–301.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany. Association for Computational Linguistics.

Pawel Swietojanski and Steve Renals. 2014. Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 171–176. IEEE.

Jörg Tiedemann. 2016. Finding alternative translations in a large corpus of movie subtitle. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

David Vilar. 2018. Learning hidden unit contribution for adapting neural machine translation models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 500–505. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas. Association for Computational Linguistics.

# Denoising Neural Machine Translation Training with Trusted Data and Online Data Selection

**Wei Wang**
Google Research
wangwe@google.com

**Taro Watanabe**
Google Research
tarow@google.com

**Macduff Hughes**
Google Research
macduff@google.com

**Tetsuji Nakagawa**
Google Research
tnaka@google.com

**Ciprian Chelba**
Google Research
ciprianchelba@google.com

## Abstract

Measuring domain relevance of data and identifying or selecting well-fit domain data for machine translation (MT) is a well-studied topic, but denoising is not yet. Denoising is concerned with a different type of data quality and tries to reduce the negative impact of data noise on MT training, in particular, neural MT (NMT) training. This paper generalizes methods for measuring and selecting data for domain MT and applies them to denoising NMT training. The proposed approach uses trusted data and a denoising curriculum realized by online data selection. Intrinsic and extrinsic evaluations of the approach show its significant effectiveness for NMT to train on data with severe noise.

## 1 Introduction

Data noise is an understudied topic in the machine translation (MT) field. Recent research has found that data noise has a bigger impact on neural machine translation (NMT) than on statistical machine translation (Khayrallah and Koehn, 2018), but learning what data quality (or noise) means in NMT and how to make NMT training robust to data noise remains an open research question.

On the other hand, a rich body of MT data research focuses on *domain data* relevance and selection for domain adaptation purpose. As a result, effective and successful methods have been published and shown to work for both SMT and NMT. For example, (Axelrod et al., 2011) introduce a metric for measuring the data relevance to a domain by using n-gram language models (LM). (van der Wees et al., 2017) employ a neural-network version of it and propose a gradually-refining strategy to dynamically schedule data during NMT training. In these methods, a large amount of in-domain data are used to help measure data domain relevance.

Data noise is a different quality that has been shown to affect NMT performance in particular. In MT, the use of web crawl, automatic methods for parallel data mining, sentence alignment provide us with parallel data of variable quality from many points of view: sentence breaking, poor sentence alignments, translations, domain adequacy, tokenization and so forth. To deal with such data noise, a commonly used practice is (static) data filtering with simple heuristics or classification. The NMT community increasingly realizes that this type of quality matters for general NMT translation accuracy. For example, (Khayrallah and Koehn, 2018) studies the types of data noise and their impact on NMT; WMT 2018 introduces a Parallel Corpus Filtering task on noisy web-crawled data.

Unfortunately, the ingredients that made domain data selection methods successful have not been studied in the NMT denoising context. Specifically,

- How to measure noise?
- How does noise dynamically interact with the training progress?
- How to denoise the model training with a small, trusted parallel dataset?

In the denoising scenario, the trusted data would be the counterpart of in-domain monolingual data of domain data selection. Trusted data can be human translations, a small amount of which can be easily available as a development set or validation set from a normal MT setup.

We use the example in Table 1 to illustrate the challenges in the NMT denoising problem, as well as the issue of directly applying existing domain methods as is for this purpose. Both sentences in the example appear to be relevant to travel conversations, but the sentence pair is “noisy” in that,

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 133–143
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64014

| | |
|---|---|
| zh | gongche zhan zai nali? |
| *zh-gloss* | *bus stop is where?* |
| en | Where is the bus stop? For bus 81. |

Table 1: A noisy sentence pair.

a part of the English sentence does not align to anything on the Chinese side, yet the pair contains some translation and the sentences are fluent. An LM-based domain-data selection method would generally treat it as a suitable domain example for building a travel NMT model and may not consider this noise.

A simple data filtering method based on length or a bilingual dictionary can easily filter it, but, intuitively, the example may still be useful for training the NMT model, especially in a data-scarce scenario – the Chinese sentence and the first half of the English sentence are still a translation pair. This suggests the subtlety in identifying noisy data for MT – It is not a simple binary problem: Some training samples may be partially useful to training a model, and their usefulness may also change as training progresses.

An NMT model alone may be incapable of identifying noise. Under a conditional seq2seq NMT model that translates Chinese into English, a word, e.g., 81, in the extra English fragment may receive a low probability (or a high loss), but that could as well mean that is hard but still correct translation. Here is then where the trusted data can play a role – It can help produce a (slightly) better model for the first model to compare against to be able to distinguish informative hard examples from harmful noisy ones.

In this paper, we propose an approach to denoising online NMT training. It uses a small amount of trusted data to help models measure noise in a sentence pair. The noise is defined based on comparison between a pair of a noisy NMT model and another, slightly denoised NMT model, inspired by the contrastive in-domain LM vs out-of-domain LM idea. It employs online data selection to sort sentence pairs by noise level so that the model is trained on gradually noise-reduced data batches. We show that language model based domain data selection method as is does not work well whereas the proposed approach is quite effective in denoising NMT training.

## 2 Related Research

One line of research that is related to our work is data selection for machine translation. It has been mostly studied in the domain adaptation context. Under this context, a popular metric to measure domain relevance of data is based on cross entropy difference (CED) between an in-domain and an out-of-domain language models. For example, (Moore and Lewis, 2010) selects LM training data with CED according to an in-domain LM and a generic LM. (Axelrod et al., 2011) propose the contrastive data selection idea to select parallel domain data. It ranks data by the bilingual CED that is computed, for each language, with a generic n-gram LM and a domain one. Even more recently, (van der Wees et al., 2017) employ a neural-network version of it along with a dynamic data selection idea and achieve better domain data selection outcome. (Mansour et al., 2011) compute the CED using IBM translation Model 1 and achieve the best domain data selection/filtering effect for SMT combined with LM selection; The case of partial or misalignments with a bilingual scoring mechanism rather than LMs is also discussed.

Another effective method to distinguish domain relevance is to build a classifier. A small amount of trusted parallel data is used in classifier training. For example, (Chen and Huang, 2016) use semi-supervised convolutional neural networks (CNNs) as LMs to select domain data. Trusted data is used to adapt the classifier/selector. (Chen et al., 2016) introduce a bilingual data selection method that uses CNNs on bitokens; The method uses parallel trusted data and is targeted at selecting data to improve SMT; In addition to domain relevance, the work also examines its noise-screening capability; The method is tried on NMT and does not seem to improve.

Previous work on domain data selection has shown that the order in which data are scheduled matters a lot for NMT training, a research that is relevant to curriculum learning (Bengio et al., 2009) in machine learning literature. (van der Wees et al., 2017) show the effectiveness of a nice “gradually-refining” dynamic data schedule. (Sajjad et al., 2017) find the usefulness of a similar idea, called model stacking for NMT domain adaptation. Data ordering could be viewed as a way of data weighting, which can be also done by example weighting/mixing, e.g., (Wang et al., 2017; Britz et al., 2017; Matsoukas et al., 2009). In the context of denoising, the quality that the ordering uses would be the amount of noise in a sentence pair, not (only) how much the data fits the domain of interest.

SMT models tend to be fairly robust to data noise and denoising in SMT seems to have been a lightly studied topic. For example, (Mediani, 2017) uses a small, clean seed corpus and designs classifier filter to identify noisy data with lexical features; and also there is a nice list of works accumulated over years, compiled on the SMT Research Survey Wiki[1].

The importance of NMT denoising has been increasingly realized. (Khayrallah and Koehn, 2018) study the impact of five types of artificial noise in parallel data on NMT training and find that NMT is less robust to data noise. (Vyas et al., 2018) select well-translated examples by identifying semantic divergences in parallel data. (Lample et al., 2017) bootstrap backtranslations with a denoising loss term, in an unsupervised NMT context. Label noise is also a generally studied topic, e.g., (Natarajan et al., 2013).

In a sense, our approach is an application of active learning (Settles, 2010). Active learning is usually employed for the model to interactively choose novel examples to obtain labels for further training a given model. In our case we use the idea to select the already labeled data that the model finds useful at a given point during training. The usefulness signal is guided by a small trusted dataset.

## 3 Online NMT Training

We usually train NMT models with online optimization, e.g., stochastic gradient descent. At a time step $t$, we have an NMT model $p(y|x;\theta_t)$ translating from sentence $x$ to $y$ with parameterization $\theta_t$. The model choice could be, for example, RNN-based (Wu et al., 2016), CNN-based (Gehring et al., 2017), Transformer model (Vaswani et al., 2017) or RNMT+ (Chen et al., 2018). To move $p(y|x;\theta_t)$ to next step, $t+1$, a *random* data batch $b_t$ is normally used to compute the cross entropy loss. The prediction accuracy of $p(y|x;\theta_t)$ does not depend on the data of this batch alone, but on all data the model has seen so far.

## 4 The Denoising Problem

The problem we address in the paper is as follows. We have a large, noisy, mixed-domain dataset $\widetilde{D}$ whose size is on the order of hundreds of millions of sentence pairs or larger. An NMT model trained on this noisy data may suffer from low translation accuracy or severe translation errors. We also have a small trusted dataset $\widehat{D}$ consisting of several thousand sentence pairs or even less. We address the denoising scenario where the trust fraction $|\widehat{D}|/|\widetilde{D}| \ll 1$ ($|\widehat{D}|$ being the size of $\widehat{D}$).

Trusted data can be human translations or any other source of parallel data of higher quality than the translations produced by our model. The trusted data we use in experiments contains noise, too. We think that, for the trusted data to improve, it needs to be stronger than the translation quality from the model we are improving, and as we will show, we define the noise level of a sentence pair relative to a model.

We are concerned with a method for selecting noise-reduced data batches to train the NMT model using online training. The trusted data is used to help measure data noise in a sentence pair. Training data is digested by training in terms of (cross entropy) loss, thus selecting noise-reduced sentence pairs to train on would be equivalent to denoising the training loss term (thus the training process).

## 5 Our Approach

We first define how to measure noise with the help of the small trusted dataset. Then we use it to control the schedule of the data batches to train the NMT model.

### 5.1 Incremental denoising with trusted data

Given a model $p(y|x;\widetilde{\theta})$ trained on noisy data $\widetilde{D}$, a practical way to denoise it with a small amount of trusted data $\widehat{D}$ would be to simply fine-tune the model on the trusted data, considering that a small trusted dataset alone is not enough to reliably train an NMT model from scratch. Fine-tuning has been used in NMT domain adaptation to adapt an existing NMT model on a small amount of in-domain data, for example, in (van der Wees et al., 2017). We hypothesize that it would be effective for denoising, too, which will be verified by our experiments.

To facilitate the introduction of our denoising method, we introduce a *denoise* function that de-

[1] http://www.statmt.org/survey/Topic/CorpusCleaning

noises a model, $p(y|x;\widetilde{\theta})$, on the trusted data $\widehat{D}$ by fune-tuning:

$$p(y|x;\widehat{\theta}) \quad = \quad \text{denoise}\left(p(y|x;\widetilde{\theta}), \widehat{D}\right) \qquad (1)$$

Eq 1 represents that model $p(y|x;\widetilde{\theta})$ with initial parameterization $\widetilde{\theta}$ is fine-tuned on the trusted data $\widehat{D}$ to yield a denoised model, $p(y|x;\widehat{\theta})$. With a small trusted dataset, the fine-tuning may take a small number of training steps.

### 5.2 Definition of data noise

MT training samples can be noisy in many ways, and different types of noise might have different impact on NMT. Furthermore, human's definition of data noise may not be completely consistent with NMT model's perspective. Therefore, instead of defining noise in these aspects, we simply use model probabilities and rely on the quality of the trusted data. After all, data needs to be ingested by model training via (cross-entropy) loss.

Supposed we are given a *noisy model*, $p(y|x,\widetilde{\theta})$, that has been trained on noisy data and a *denoised model*, $p(y|x,\widehat{\theta})$, obtained by Eq 1, with the denoised model being a slightly more accurate probability distribution than the noisy version. For a sentence pair $(x, y)$ of a source sentence $x$ and its target translation $y$, we can compute its "noisy logprob" under the noisy model:

$$L_{p(y|x,\widetilde{\theta})} \quad = \quad \log p(y|x,\widetilde{\theta}) \qquad (2)$$

We can also compute its "denoised logprob" under the denoised model:

$$L_{p(y|x,\widehat{\theta})} \quad = \quad \log p(y|x,\widehat{\theta}) \qquad (3)$$

We then define the noise level of a sentence pair $(x, y)$ as the difference of a noisy model score over a denoised model score:

$$\text{noise}(x, y; \widetilde{\theta}, \widehat{\theta}) \quad = \quad L_{p(y|x;\widetilde{\theta}_t)} - L_{p(y|x;\widehat{\theta}_t)} \qquad (4)$$

The noise level of a sentence pair is the sum of the per-word noise over all the target words (under conditional translation models). Noise$(x, y; \widetilde{\theta}, \widehat{\theta})$ could also be normalized by the length of sentence $y$ empirically. The bigger noise$(x, y; \widetilde{\theta}, \widehat{\theta})$ is, the higher noise level the sentence pair has. A negative value of noise$(x, y; \widetilde{\theta}, \widehat{\theta})$ means that the sentence pair has more information to learn from (cleaner).

The noise in a sentence pair is defined in terms of the comparison between two models: the noisy model, $\widetilde{\theta}$, and the denoised model, $\widehat{\theta}$. Under this definition, noise is relative – A sentence pair could have negative noise$(x, y; \widetilde{\theta}, \widehat{\theta})$ (not noise) for weeker models (i.e., an earlier checkpoint of $\widetilde{\theta}$ in an NMT training), but could become noisy (positive value) for stronger models (i.e., a later checkpoint of $\widetilde{\theta}$). This would address one of the issues we illustrated in Section 1 with the example in Table 1.

Notice that this definition of noise is a generalization of the bilingual cross-entropy difference (CED) defined and used in (Axelrod et al., 2011; van der Wees et al., 2017) to measure domain relevance of a sentence pair. We use seq2seq NMT models to directly model a sentence pair, while previous works use language models to model monolingual sentences independently. A language model corresponds just to the decoder component of a translation model and thus cannot model the translation quality. The lack of the encoder component (thus translation) makes the LM-based method unsuitable for denoising, as we show in experiments. Additionally, we use a small, bilingual trusted dataset (semi-supervision) rather than lots of in-domain data (heavier supervision).

### 5.3 Denoising by online data selection

#### 5.3.1 The idea

Our main idea for online denoising of NMT training is to train an NMT model on a progressively-denoised curriculum (data batches). As a result, the entire training becomes a continuous fine-tuning. We realize the denoising curriculum through dynamic data selection to "anneal" the noise level in data batches over training steps. Therefore, our method tries to control the way how noise dynamically interacts with training loss by data selection, instead of directly altering per-example loss. The assumption is that $\widetilde{D}$ contains good examples to select, which is usally true with a big enough training dataset $\widetilde{D}$.

More concretely, at each step with an initial (potentially still noisy) model, $p(y|x;\widetilde{\theta}_t)$, the method denoises it (by Eq 1) with the trusted data $\widehat{D}$ into a slightly better model $p(y|x;\widehat{\theta}_t)$ for that step. With this pair of noisy and denoised models, we then compute noise scores for examples in a buffer $\widetilde{B}_t^{\text{random}}$ that is randomly drawn from $\widetilde{D}$ per step and maintained during training. We sort the noise

**Algorithm 1:** Denoising NMT training with trusted data and online data selection.

```
Input: Noisy data $\widetilde{D}$, trusted data $\widehat{D}$
Output: A denoised, better model
$t = 0$; Randomly initialize $\widetilde{\theta}_0$.
while $t < T$ do
    $p(y|x; \widehat{\theta}_t) \leftarrow$ denoise($p(y|x; \widetilde{\theta}_t)$, $\widehat{D}$).
    Randomly draw $\widetilde{B}_t^{\text{random}}$ from $\widetilde{D}$.
    Compute noise for examples in $\widetilde{B}_t^{\text{random}}$ by Eq 4.
    Sort $\widetilde{B}_t^{\text{random}}$ by noise scores.
    Sample $b_t$ from top $r_t$ of above sorted buffer.
    Train $p(y|x; \widetilde{\theta}_t)$ on $b_t$ to produce new model $p(y|x; \widetilde{\theta}_{t+1})$.
    Discard the denoised model $p(y|x; \widehat{\theta}_t)$.
    $t \leftarrow t + 1$.
end while
```

scores. The final, actual data batch $b_t$ is then randomly sampled from the top $r_t$ portion of $B_t^{\text{random}}$ based on the sorted scores, where $r_t$, called *selection ratio*, is increasingly tightened. Averaged noise level of examples in the top $r_t$ portion expects to become less over time. As a result, the data batches $b_t$'s that are actually fed to train the final model are gradually denoised. Algorithm 1 summarizes the idea. It is worth pointing out that this denoising method is realized by a bootstrapping process, in which, $\widetilde{\theta}_t$ and $\widehat{\theta}_t$ iteratively bootstrap each other by interacting with the trusted data and selected denoised data.

We choose to use the following exponential decaying function for selection ratio, $r_t$, to anneal data noise by data selection[2]:

$$r_t = 0.5^{t/H} \tag{5}$$

It keeps decreasing/tightening over time $t$. The entire training thus becomes a continuous fine-tuning process, in a self-paced learning (Kumar et al., 2010) fashion.

In Equation 5, $H$ is a hyper-parameter controlling the decaying pace: It halves $r_t$ every $H$ steps. For instance, $H = 10^6$ means that, at step 1 million, data batch $b_t$ will be drawn from the top-50% out of sorted buffer.

In practice, it may be desirable to set a floor value for $r_t$ (e.g., 0.2) to avoid potential selection bias. $B_t^{\text{random}}$ needs also to be big enough such that there are enough examples in the top $r_t$ range to select from to form the final training batch $b_t$, which is usally a constant size – It needs to contain at least $|b_t|/r_{\text{floor}}$ examples.

The noise annealing is inspired by (van der Wees et al., 2017), but we anneal data quality at per step to make the approach more friendly to NMT online optimization, instead of per data epoch. Compared to static selection, the noise annealing idea also makes every training example useful, by digesting noisy examples earlier and fine-tuning on good-quality examples later on.

Note that there are two reasons that this process does not overfit on the trusted data, even though it is kept being used to denoise the initial model at every step. First, the noisy model, $p(y|x; \widetilde{\theta})$ being trained over steps is never trained on the trusted data – It is the denoised model, $p(y|x; \widehat{\theta})$, that is trained on it and then gets discarded at the end of that step. Second, the online data selection progressively anneals from noisy examples to less noisy ones, instead of greedily keeping selecting out of the least noisy examples.

### 5.3.2 Data selection per-step overhead

Compared to normal NMT training, there is a per-step data selection overhead in Algorithm 1. The overhead includes (1) training the denoised model on a small trusted dataset, which requires a small number of training steps; and (2) scoring all examples in the random buffer $B_t^{\text{random}}$ with both the noisy model and the denoised model. Both cases will in general depend on model size, but will probably depend even more on model type and configuration.

### 5.3.3 Lightweight implementation

We make Algorithm 1 more lightweight by decoupling model training from example noise scoring: We score all examples in $\widetilde{D}$ offline and use scores for online data selection.

Algorithm 2 shows the details of this idea. To enable offline scoring, we train the noisy model and the denoised model prior to the final, target training, on the noisy data $\widetilde{D}$ and the trusted data $\widehat{D}$, respectively. We then use this pair of models to score all examples in $\widetilde{D}$ and save the scores. In target model training, the example are retrieved into the buffer with scores, without the need of com-

[2] We simply use one of the ways to anneal learning rate as the decaying function to anneal training data selection.

**Algorithm 2:** Lightweight implementation of Algorithm 1. Actually used in experiments.

```
Input: Noisy data $\widetilde{D}$, trusted data $\widehat{D}$
Output: A denoised, better model with learned parameters $\Theta$.
Train $p(y|x;\widetilde{\theta})$ with small $\widetilde{\theta}$ on $\widetilde{D}$.
$p(y|x;\widehat{\theta}) \leftarrow$ denoise $\left(p(y|x;\widetilde{\theta}); \hat{D}\right)$.
Score $\widetilde{D}$ with $\widetilde{\theta}$ and $\widehat{\theta}$ by Eq 4.
$t = 0$; Randomly initialize $\widetilde{\Theta}_0$.
while $t < T$ do
    Randomly sample $\widetilde{B}_t^{\text{random}}$ from $\widetilde{D}$.
    Sort $\widetilde{B}_t^{\text{random}}$ by offline-computed noise scores.
    Sample $b_t$ from top $r_t$ of above sorted buffer.
    Train $p(y|x;\widetilde{\Theta}_t)$ on $b_t$ to produce new model $p(y|x;\widetilde{\Theta}_{t+1})$.
    $t \leftarrow t + 1$.
end while
```

puting on the fly. Then the remaining is similar to Algorithm 1. This effectively turns the per-step data selection overhead in Algorithm 1 into constant overhead.

We can also use smaller parameterization for the noisy model and denoised model than the target model. This may not affect their noise-discerning capability as long as they are still seq2seq models, the same as the target model. This is because we define the noise score in terms of logprob difference and use the scores for ranking/selection (e.g., via top $r_t$),

In summary, here is the lightweight method that we eventually use to denoise NMT training with trusted data and online data selection: Train $p(y|x;\widetilde{\theta})$ on noisy data $\widetilde{D}$ with a small parametrization. Denoise $p(y|x;\widetilde{\theta})$ on trusted data $\widehat{D}$ to produce denoised model $p(y|x;\widehat{\theta})$ (Eq 1). Score the entire noisy data $\widetilde{D}$ with the above two models by Eq 4. Train the target model with the above online, dynamic data selection. Algorithm 2 describes the idea.

We are going to use this implementation in experiments. Note, however, that we find that the general method in Algorithm 1 is very useful in understanding the nature of the denoising problem and thus cannot be ignored in the context. For example, it makes us realize the denoising problem is about how to (actively) meet what the model needs, i.e., not standalone filtering. And also, the bootstrapping behavior in Algorithm 1 further motivates the use of the noise-annealing online data selection strategy and helps refine the lightweight implementation.

## 6 Experiments

### 6.1 Setup

We carry out experiments for en/fr with two training datasets ($\widetilde{D}$), respectively. Paracrawl[3] en/fr training raw data has 4 billion sentence pairs. After removing identities and empty source/target, about 300 million (M) sentence pairs are left. WMT 2014 en/fr training data has about 36M sentence pairs, with provided sentence alignment.

WMT newstest 2012-2013 is used as the development set for early stopping of training. We use three test sets: WMT (n)ewstest 2014 (**n2014**), news (d)iscussion test 2015 (**d2015**), and a 2000-line patent test set (**patent**)[4]. More test sets than just n2014 are used in order to confirm that the gain obtained is not only from news domain adaptation but cross-domain, general accuracy improvement.

The WMT newstest 2010-2011 is used as the trusted data. It contains 5500 sentence pairs. We acknowledge that ideal trusted data would probably be both well-translated and domain-matched, but we leave the study of trusted data properties to future research.

We compute the *detokenized* and mixed-cased BLEU scores against the original references (per (Post, 2018)) with an in-house implementation of script `mteval-v14.pl`.

We use an RNN-based NMT architecture similar to (Wu et al., 2016). Our final model has 8 layers of encoder/decoder, 1024 dimensions with 512-dimension attention. The smaller selector (noisy and denoised) models (in Algorithm 2) are of 3 layers and 512 dimensions.[5]

Denoising a model on the small trusted dataset is done by fine-tuning on it by SGD. The training is terminated with early stopping by checking the perplexity on the development set. It is a tiny dataset, but as we will show, its denoising impact is quite impressive and surprising. Training on

[3] `http://statmt.org/paracrawl`

[4] Obtained from `https://www.epo.org`.

[5] Even smaller models like 2-layer x 256-dimension works, too, when we examined on an internal dataset.

such a small data can easily overfit, we thus use a very small learning rate 5e-5 so that the training progresses slow enough for us to reliably catch a good checkpoint before training stops.

In Paracrawl trainings, we train for 3M steps using SGD with learning rate 0.5 and start to anneal/reduce the learning rate at step 2M by halving it every 200k steps. In WMT training, we train for 2M steps with learning rate 0.5 but start to anneal learning rate at step 1.2M with the same pace. We use dropout 0.2 for the WMT training. We did not use dropout for Paracrawl training due to its large training data amount.

To dynamically anneal the data batch quality (Eq 5), we set hyper-parameter $H$ to step 1.1M. 0.2 is used as the floor selection ratio, $r_t$. The rationale for the choice of $H$ is so that when learning rate annealing happens, $r_t$ is close to its minimum value to ensure the training is indeed trained on the desired, best selected data.

## 6.2 Training data cleanness

To measure how noisy the datasets are, we randomly sample 2000 sentence pairs from the WMT dataset. Human raters were asked to label each sentence pair with scales in Table 2.

These ratings generally reflect how well-translated a sentence pair is, however, a rating 4 does not necessarily mean that is exactly the type of data a model needs – Model's perspective on good data may not completely consistent with human, because these ratings are not necessarily connected to data loss of a model. We use these ratings mainly to assess if our noise definition correlates to these ratings to some extent, but the noise definition could do more. The rater agreement on good ($>=$ 3) or bad ($<$ 3) is 70% and we find the averaged rating is very reliable and stable to measure a small sentence pair sample.

Table 3 shows that WMT 2014 data is relatively clean: it has 40% rated as perfect; its averaged rating is 3.0 (4 being perfect). Noise introduced by sentence alignment accounts for part of the low ratings. We did not rate a Paracrawl sample, since just eyeballing a sample of the data reveals that it was noisy consisting of many boilerplates, wrong language identification, wrong translations.

## 6.3 Noise score vs human rating

We expect the noise definition (Eq 4) to correlate with the averaged cleanness of selected data and the dynamic scheduling method schedules data from noisy to clean. We verify this on the sample with human ratings.

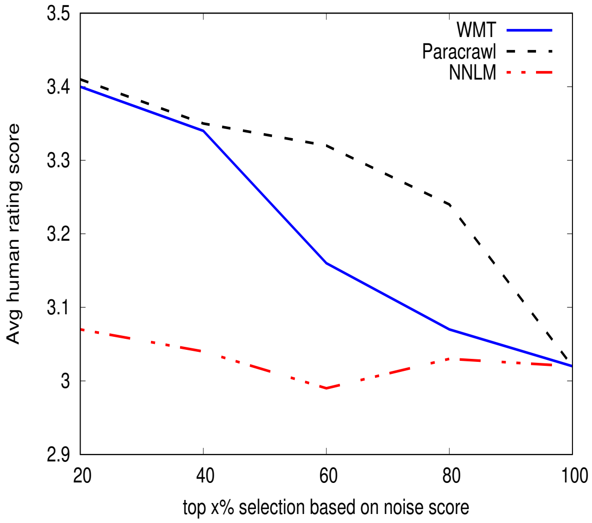


Figure 1: Noise-discerning capability of different noise scoring models. Curves are drawn by selecting, according to Eq 4, top $x\%$ (x-axis) out of a rated sample of 2000 random sentence pairs from the WMT en/fr dataset. WMT: noise scoring models trained on WMT training data, and trusted data. Paracrawl: noise scoring models trained on Paracrawl data, and trusted data. NNLM: neural net based LM selection models trained on Paracrawl data, and trusted data. Trusted data are the same dataset.

We carry out steps 1 and 2 of the practical implementation in Section 5.3.3 to produce the small noisy model and its denoised model. Recall that they are used to compute the noise in each sentence pair by Eq 4. We repeat this on the Paracrawl data and the WMT data, respectively, and thus we have two pairs of models, one for each dataset.

We apply each pair of models to score the rated WMT sample, sort the sentence pairs by noise scores. We then select $x\%$ least noisy sentence pairs. Each $x\%$ corresponds to a subset and we compute the averaged human rating for that subset. In Figure 1, x-axis shows $x\%$, the percentage out of the entire sample; y-axis shows the averaged human rating for the $x\%$ selection subset. Going from right to left, data indeed becomes cleaner as selection becomes tighter for the scoring models in our proposed method: WMT is noise scoring models trained on WMT training data, and trusted data. Paracrawl is the noise scoring models trained on Paracrawl data, and trusted data. Trusted data are the same dataset. We explain the dot-dashed line in a later experiment (Section 6.6).

Ranking capability of the Paracrawl selector

| Rating scale | Explanation |
|---|---|
| 4 (Perfect) | Almost all information (90-100%) in the sentences is conveyed in each other. |
| 3 (Good) | Most information (70-90%) in the sentences is conveyed. |
| 2 (Not good) | Some (30-70%) information in the sentences is conveyed, but some is not. |
| 1 (Bad) | (10-30%) A large amount of information in the sentences is lost or misinterpreted. |
| 0 (Poor) | (0-10%) The two sentences are nearly or completely unrelated, or in wrong languages. |

Table 2: Scales for human rating sentence pairs. Percentage ranges refer to the amount of words well translated across sentences in a pair.

| Rating scale | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| WMT | 47% | 31% | 10% | 3% | 9% |

Table 3: Rating stats on an en/fr WMT training data sample of 2000 sentence pairs.

| System | n2014 | d2015 | patent |
|---|---|---|---|
| Paracrawl dataset | | | |
| P1 Paracrawl baseline | 31.6 | 30.7 | 37.2 |
| P2 Incr-denoise P1 | 34.0 | 33.7 | 44.7 |
| P3 Online denoised | **35.2** | **35.6** | **46.9** |
| WMT dataset | | | |
| W1 WMT baseline | 36.2 | 35.8 | 45.7 |
| W2 Incr-denoise W1 | 36.2 | 35.8 | 45.7 |
| W3 Online denoised | **36.9** | **36.4** | **46.1** |

Table 4: BLEU scores of Denoising experiments with en/fr Paracrawl data and WMT data. "Incr-denoise P1" refers to applying the incremental denoising on the noisy baseline P1 with method in Section 5.1. Similarly for "incr-denoise W1". Under paired bootstrapped test at $p < 0.05$, P3 is significantly better than P2, P3 than P1, P2 than P1, on all test sets. W3 is significantly better than W1 on n2014.

seems slightly better than the WMT one in discerning noisier sentence pairs. We speculate this is because the noisy Paracrawl data "amplifies" the contrastive effect of the pair of models.

### 6.4 BLEU scores

BLEU scores in Table 4 show the impact of denoising. For each training dataset, we have three experiments: baseline, noisy training with random data batch selection (P1 and W1); Denoising baseline with trusted data by fine-tuning the baseline on it (Eq 1) (P2 and W2); Training a model from scratch with online training by dynamic, gradually noise-reduced data selection (P3 and W3).

First of all, P1 vs P2, it is impressive that just fine-tuning a noisy baseline on a small trusted dataset yiels a big impact. P2 improves P1 by +2.4 BLEU on n2014, +3 BLEU on d2015 and +7.5 BLEU on patent. The Paracrawl experiments and the above rating ranking curves (Figure 1) indicate the power of simple incremental denoising on trusted data (Section 5.1) when the background data is very noisy. In NMT domain adaptation literature (e.g., (van der Wees et al., 2017)), it is known that fine-tuning on domain data improves domain test sets, but it is also known that it may hurt test sets that are out of domain (forgetting). We think our experiments are the first to report the incremental denoising power of fine-tuning on a tiny trusted data. Notice incremental denoising does not improve on WMT data (W1 vs W2) probably because WMT data is relatively cleaner. This, however, would indicate that the gain for P1 vs P2 is less likely a domain adaptation effect.

P2 vs P3 shows that the online denoising approach reduced the training noise further more and gains +1.2 n2014 BLEU, +1.9 d2015 BLEU and +2.2 patent BLEU, on top of incremental denoising on trusted data. On the WMT dataset, W2 vs W3 shows that, even though the trusted data does not directly help, the online denoising helps by +0.7 n2014 BLEU, +0.6 d2015 BLEU and +0.4 patent BLEU. We carried out paired bootstrapped statistical significance test (Koehn, 2004) between systems, at $p < 0.05$, P3 is significantly better than P2, P3 than P1, P2 than P1, across all test sets; W3 is significantly better than W1 only on n2014.

We also would like to note the strength of the WMT baseline system (W1). Its n2014 BLEU is 36.2, detokenized, case-sensitive. Published literatures tend to report tokenized, case-sensitive BLEU scores, for which W1 BLEU becomes 40.2 on the same test set. This is a strong score with a standard LSTM RNN network, compared to published results for this task.

| System | n2014 | d2015 | patent |
|---|---|---|---|
| Paracrawl dataset | | | |
| P1 Random order | 31.6 | 30.7 | 37.2 |
| P3 Online denoised | **35.2** | **35.6** | **46.9** |
| P4 Reverse order of P3 | 32.6 | 31.1 | 40.9 |

Table 5: Online denoising: NMT trained on data sorted according to noisiness level. P3 is trained on noisier to cleaner data order. Reversely, P4 is trained on cleaner to noisier data order.

.

| | Subset | n2014 | d2015 | patent |
|---|---|---|---|---|
| P1 | | 31.6 | 30.7 | 37.2 |
| P5 | $S_{80\%}$ | 33.1 | 32.3 | 44.3 |
| P6 | $S_{40\%}$ | 33.9 | 34.4 | 45.1 |
| P7 | $S_{20\%}$ | 34.4 | 34.6 | 45.6 |

Table 6: Nested datasets: Data order is important for denoising. $S_{80\%} \supset S_{40\%} \supset S_{20\%}$ with stricter/smaller set less noisy.

### 6.5 Data order

Our online denoising method dynamically selects data batches whose noise is gradually reduced to train the target model. We carry out two sets of experiments to prove that this is necessary for denoising.

In the first experiment, we compare P3 (in proposed data order) to the "reverse" of P3, where data batches are dynamically scheduled in a reverse, noise gradually increasing order such that the model is trained on cleaner data earlier and then noisier data later (i.e., by simply flipping the sign of Eq 4) – The entire training then becomes a continuous reverse fine-tuning. Table 5 shows that the reverse order (P4) clearly does not work as effective for denoising, even though P4 still slightly improves the baseline with random data selection (P1 in Table 4).

In another experiment, we select 3 data subsets based on the amount of noise in each sentence pair, each subset being noise-reduced to different degree. For example, we select top 80% least noisy sentence pairs (denoted as $S_{80\%}$) out the entire Paracrawl data. Then we select the top half of $S_{80\%}$ which is essentially $40\%$ of the Paracrawl data. We denote it as $S_{40\%}$, similarly, $S_{20\%}$, therefore $S_{80\%} \supset S_{40\%} \supset S_{20\%}$. And we expect the averaged noise in the smaller percentage would be less according to Figure 1. Then we fine-tune P1 (noisy baseline) on $S_{80\%}$ with early stopping on devset, followed by the fine-tuning on $S_{40\%}$ and then $S_{20\%}$. Table 6 shows that each stricter subset is able to boost the previous training across all test sets, by further denoising. This also confirms the importance of the right data order in denoising.

P3 vs P4 seems to confirm the spirit of Curriculum Learning (Bengio et al., 2009) – CL promotes ordering data to gradually focus on those most important examples, and here the training has a better outcome (P3) by training on progressively noised-reduced data.

### 6.6 Language model selection

The proposed method uses seq2seq NMT models for online data selection. We can replace them with neural network language models (NNLM) with everything else the same, to confirm that the LM based method that is popular for domain data selection is not designed for denoising.

We first check if the NNLM selection scores correlate with human ratings. As shown by the dot-dashed line (red) in Figure 1, it does not seem to – As we tighten the selection percentage (from right to left), the averaged rating of sentence pairs falling into that percentage does not increase, but the method that employs the seq2seq models to compute noise scores (Eq 4) does.

We also compare the BLEU scores of the NNLM selection and the NMT selection. To that end, we select top $20\%$ data and use it to fine-tune the noisy Paracrawl baseline (P1), for the NNLM method and the proposed method, respectively.

We had to resolve an issue in the NNLM selection experiment. Recall that the trusted data we use is from WMT newstest 2010-2011 and the development set we use for stopping the training is WMT newstest 2012-2013. WMT newstests across years do not seem to be in the same domain, as a result, the perplexity on devset never drops in training with trusted data. This would be additional evidence that improvements from our proposed denoising approach is unlikely from domain adaptation. In the end, we had to extract randomly 1000 lines out of the trusted data as the devset for early stopping and use the remaining as the trusted data when training the denoised model $\widehat{\theta}$ that is used to compute the noise scores (or data relevance in the NNLM case) by Eq 4.

The BLEU scores in Table 7 show the clear difference. The NNLM method does not discern noise and thus the top selection would be

| System | n2014 | d2015 | patent |
|---|---|---|---|
| P1 Paracrawl baseline | 31.6 | 30.7 | 37.2 |
| P8 P1+NMT 20% | **34.3** | **34.7** | **45.8** |
| P9 P1+NNLM 20% | 31.8 | 30.5 | 35.4 |

Table 7: LM method does not denoise, but NMT method (proposed) does; and a denoised model has improved general translation accuracy. P1+NMT 20%: fine-tune P1 with top 20% selection by NMT method. P1+NNLM 20%: fine-tune P1 with top 20% selection by NNLM method.

as noisy as the baseline data. As a result, fine-tuning the noisy baseline (P1) would not improve. As a matter of fact, the patent BLEU drops over baseline, probably indicating that domain data selection causes data bias. The proposed method, on the other hand, performs clearly better (P8), for example, compared to P9, +2.5 BLEU on n2014, +3.8 BLEU on d2015 and +10.4 BLEU on patent. These prove the effectiveness of the proposed method in producing better systems on noisy data.

### 6.7 Discussion

The research in (van der Wees et al., 2017) that selects data with neural language models show that dynamically selected parallel data for domain adaptation improves domain test sets, but it can hurt test sets that are out of domain. It also shows that the dynamic online selection still underperforms the fine-tuning on domain parallel data. In our denoising results, the online denoising (e.g., P3) can significantly outperform the simple fine-tuning (e.g., P2).

We clarify that our method could potentially work with other data filtering methods. For example, if the underlying noisy data has already been filtered, applying online denoising with trusted data could potentially bring even further improvement than no pre-filtering.

## 7 Conclusion and Future Research

Domain data selection and domain adaptation for machine translation is a well-studied topic, but denoising training data or MT training is not yet, especially for NMT training. In this paper, we generalize the recipes of effective domain data research to address a different and important data quality for NMT – data noise. We define how to measure noise and how to select noise-reduced data batches to train NMT models online. We show that the noise we define correlates with human ratings and that the proposed approach yields significantly better NMT models.

The method probably can be tried to denoising for other seq2seq tasks like parsing, image labeling. It seems interesting to study and understand the properties that trusted data should have. It also sounds an interesting research to discover better data orders.

## Acknowledgments

The authors would like to thank George Foster for his help refine the paper and advice on various technical isses in the paper, Thorsten Brants for his earlier work on the topic, Christian Buck for his help with the Paracrawl data, Yuan Cao for his valuable comments and suggestions on the paper, and the anonymous reviewers for their constructive reviews.

## References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26 th International Conference on Machine Learning*, page 86–96, Montreal, Canada.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126. Association for Computational Linguistics.

Boxing Chen and Fei Huang. 2016. Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 314–323.

Boxing Chen, Roland Kuhn, George Foster, Colin Cherry, and Fei Huang. 2016. Bilingual methods for adaptive training data selection for machine translation. In *AMTA*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *ACL 2018*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *CoRR*, abs/1805.12282.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

M. P. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1189–1197. Curran Associates, Inc.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

Saab Mansour, Joern Wuebker, and Hermann Ney. 2011. Combining translation and language model scoring for domain-specific data filtering. In *International Workshop on Spoken Language Translation*, pages 222–229.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 708–717, Singapore. Association for Computational Linguistics.

Mohammed Mediani. 2017. *Learning from Noisy Data in Statistical Machine Translation*. Ph.D. thesis, Fakultät für Informatik, Karlsruhe Institute of Technologie (KIT).

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference*, pages 220–224.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1196–1204. Curran Associates, Inc.

Matt Post. 2018. A call for clarity in reporting bleu scores. *Computing Research Repository*, arXiv:1804.08771v1. Version 2.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. 2017. Neural machine translation training in a multi-domain scenario. *arXiv preprint arXiv:1708.08712v2*.

Burr Settles. 2010. Active learning literature survey. Technical report, University of Wisconsin–Madison.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, , and Illia Polosukhin. 2017. Attention is all you need. In *CoRR abs/1706.03762*.

Yogarshi Vyas, Xing Niu, and Marine Carpuat. 2018. Identifying semantic divergences in parallel text without annotations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 1503–1515. Association for Computational Linguistics.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488. Association for Computational Linguistics.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine transaltion. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

# Using Monolingual Data in Neural Machine Translation: a Systematic Study

**Franck Burlot**
Lingua Custodia
1, Place Charles de Gaulle
78180 Montigny-le-Bretonneux
franck.burlot@linguacustodia.com

**François Yvon**
LIMSI, CNRS, Université Paris Saclay
Campus Universitaire d'Orsay
F-91 403 Orsay Cédex
francois.yvon@limsi.fr

## Abstract

Neural Machine Translation (MT) has radically changed the way systems are developed. A major difference with the previous generation (Phrase-Based MT) is the way monolingual target data, which often abounds, is used in these two paradigms. While Phrase-Based MT can seamlessly integrate very large language models trained on billions of sentences, the best option for Neural MT developers seems to be the generation of artificial parallel data through *back-translation* - a technique that fails to fully take advantage of existing datasets. In this paper, we conduct a systematic study of back-translation, comparing alternative uses of monolingual data, as well as multiple data generation procedures. Our findings confirm that back-translation is very effective and give new explanations as to why this is the case. We also introduce new data simulation techniques that are almost as effective, yet much cheaper to implement.

## 1 Introduction

The new generation of Neural Machine Translation (NMT) systems is known to be extremely data hungry (Koehn and Knowles, 2017). Yet, most existing NMT training pipelines fail to fully take advantage of the very large volume of monolingual source and/or parallel data that is often available. Making a better use of data is particularly critical in domain adaptation scenarios, where parallel adaptation data is usually assumed to be small in comparison to out-of-domain parallel data, or to in-domain monolingual texts. This situation sharply contrasts with the previous generation of statistical MT engines (Koehn, 2010), which could seamlessly integrate very large amounts of non-parallel documents, usually with a large positive effect on translation quality.

Such observations have been made repeatedly and have led to many innovative techniques to integrate monolingual data in NMT, that we review shortly. The most successful approach to date is the proposal of Sennrich et al. (2016a), who use monolingual target texts to generate artificial parallel data via backward translation (BT). This technique has since proven effective in many subsequent studies. It is however very computationally costly, typically requiring to translate large sets of data. Determining the "right" amount (and quality) of BT data is another open issue, but we observe that experiments reported in the literature only use a subset of the available monolingual resources. This suggests that standard recipes for BT might be sub-optimal.

This paper aims to better understand the strengths and weaknesses of BT and to design more principled techniques to improve its effects. More specifically, we seek to answer the following questions: since there are many ways to generate pseudo parallel corpora, how important is the quality of this data for MT performance? Which properties of back-translated sentences actually matter for MT quality? Does BT act as some kind of regularizer (Domhan and Hieber, 2017)? Can BT be efficiently simulated? Does BT data play the same role as a target-side language modeling, or are they complementary? BT is often used for domain adaptation: can the effect of having more in-domain data be sorted out from the mere increase of training material (Sennrich et al., 2016a)? For studies related to the impact of varying the size of BT data, we refer the readers to the recent work of Poncelas et al. (2018).

To answer these questions, we have reimplemented several strategies to use monolingual data in NMT and have run experiments on two language pairs in a very controlled setting (see § 2). Our main results (see § 4 and § 5) suggest promising directions for efficient domain adaptation with cheaper techniques than conventional BT.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 144–155
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64015

|  | Out-of-domain |  | In-domain |  |
|---|---|---|---|---|
|  | Sents | Token | Sents | Token |
| **en-fr** | 4.0M | 86.8M/97.8M | 1.9M | 46.0M/50.6M |
| **en-de** | 4.1M | 84.5M/77.8M | 1.8M | 45.5M/43.4M |

Table 1: Size of parallel corpora

# 2 Experimental Setup

## 2.1 In-domain and out-of-domain data

We are mostly interested with the following training scenario: a large out-of-domain parallel corpus, and limited monolingual in-domain data. We focus here on the *Europarl* domain, for which we have ample data in several languages, and use as in-domain training data the Europarl corpus[1] (Koehn, 2005) for two translation directions: English→German and English→French. As we study the benefits of monolingual data, most of our experiments only use the target side of this corpus. The rationale for choosing this domain is to (i) to perform large scale comparisons of synthetic and natural parallel corpora; (ii) to study the effect of BT in a well-defined domain-adaptation scenario. For both language pairs, we use the Europarl tests from 2007 and 2008[2] for evaluation purposes, keeping test 2006 for development. When measuring out-of-domain performance, we will use the WMT newstest 2014.

## 2.2 NMT setups and performance

Our baseline NMT system implements the attentional encoder-decoder approach (Cho et al., 2014; Bahdanau et al., 2015) as implemented in Nematus (Sennrich et al., 2017) on 4 million out-of-domain parallel sentences. For French we use samples from News-Commentary-11 and Wikipedia from WMT 2014 shared translation task, as well as the Multi-UN (Eisele and Chen, 2010) and EU-Bookshop (Skadiņš et al., 2014) corpora. For German, we use samples from News-Commentary-11, Rapid, Common-Crawl (WMT 2017) and Multi-UN (see table 1). Bilingual BPE units (Sennrich et al., 2016b) are learned with 50k merge operations, yielding vocabularies of about respectively 32k and 36k for English→French and 32k and 44k for English→German.

Both systems use 512-dimensional word embeddings and a single hidden layer with 1024 cells. They are optimized using Adam (Kingma and Ba, 2014) and early stopped according to the validation performance. Training lasted for about three weeks on an Nvidia K80 GPU card.

Systems generating back-translated data are trained using the same out-of-domain corpus, where we simply exchange the source and target sides. They are further documented in § 3.1.

For the sake of comparison, we also train a system that has access to a large batch of in-domain parallel data following the strategy often referred to as "fine-tuning": upon convergence of the baseline model, we resume training with a 2M sentence in-domain corpus mixed with an equal amount of randomly selected out-of-domain natural sentences, with the same architecture and training parameters, running validation every 2000 updates with a patience of 10. Since BPE units are selected based only on the out-of-domain statistics, fine-tuning is performed on sentences that are slightly longer (ie. they contain more units) than for the initial training. This system defines an upper-bound of the translation performance and is denoted below as `natural`.

Our baseline and topline results are in Table 2, where we measure translation performance using BLEU (Papineni et al., 2002), BEER (Stanojević and Sima'an, 2014) (higher is better) and characTER (Wang et al., 2016) (smaller is better). As they are trained from much smaller amounts of data than current systems, these baselines are not quite competitive to today's best system, but still represent serious baselines for these datasets. Given our setups, fine-tuning with in-domain natural data improves BLEU by almost 4 points for both translation directions on in-domain tests; it also improves, albeit by a smaller margin, the BLEU score of the out-of-domain tests.

# 3 Using artificial parallel data in NMT

A simple way to use monolingual data in MT is to turn it into synthetic parallel data and let the training procedure run as usual (Bojar and Tamchyna, 2011). In this section, we explore various ways to implement this strategy. We first reproduce results of Sennrich et al. (2016a) with BT of various qualities, that we then analyze thoroughly.

## 3.1 The quality of Back-Translation

### 3.1.1 Setups

BT requires the availability of an MT system in the reverse translation direction. We consider here

[1] Version 7, see www.statmt.org/europarl/.

[2] www.statmt.org/wmt08.

| | English→French | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | test-07 | | | test-08 | | | newstest-14 | | |
| | BLEU | BEER | CTER | BLEU | BEER | CTER | BLEU | BEER | CTER |
| Baseline | 31.25 | 62.14 | 51.89 | 32.17 | 62.35 | 50.79 | 33.06 | 61.97 | 48.56 |
| backtrans-bad | 31.55 | 62.39 | 51.50 | 31.89 | 62.23 | 51.73 | 31.99 | 61.59 | 48.86 |
| backtrans-good | 32.99 | 63.43 | 49.58 | 33.25 | 63.08 | 49.29 | 33.52 | 62.62 | 47.23 |
| backtrans-nmt | 33.30 | 63.33 | 50.02 | 33.39 | 63.09 | 49.48 | 34.11 | 62.76 | 46.94 |
| fwdtrans-nmt | 31.93 | 62.55 | 50.84 | 32.62 | 62.66 | 49.83 | 33.56 | 62.44 | 47.65 |
| backfwdtrans-nmt | 33.09 | 63.19 | 50.08 | 33.70 | 63.25 | 48.83 | 34.00 | 62.76 | 47.22 |
| natural | 35.10 | 64.71 | 48.33 | 35.29 | 64.52 | 48.26 | 34.96 | 63.08 | 46.67 |
| | English→German | | | | | | | | |
| | test-07 | | | test-08 | | | newstest-14 | | |
| | BLEU | BEER | CTER | BLEU | BEER | CTER | BLEU | BEER | CTER |
| Baseline | 21.36 | 57.08 | 63.32 | 21.27 | 57.11 | 60.67 | 22.49 | 57.79 | 55.64 |
| backtrans-bad | 21.84 | 57.85 | 61.24 | 21.04 | 57.44 | 59.77 | 22.28 | 57.70 | 55.49 |
| backtrans-good | 23.33 | 59.03 | 58.84 | 23.11 | 57.14 | 57.14 | 22.87 | 58.09 | 54.91 |
| backtrans-nmt | 23.00 | 59.12 | 58.31 | 23.10 | 58.85 | 56.67 | 22.91 | 58.12 | 54.67 |
| fwdtrans-nmt | 21.97 | 57.46 | 61.99 | 21.89 | 57.53 | 59.71 | 22.52 | 57.93 | 55.13 |
| backfwdtrans-nmt | 22.99 | 58.37 | 60.45 | 22.82 | 58.14 | 58.80 | 23.04 | 58.17 | 54.96 |
| natural | 26.74 | 61.14 | 56.19 | 26.16 | 60.64 | 54.76 | 23.84 | 58.64 | 54.23 |

Table 2: Performance *wrt.* different BT qualities

| | French→English | | | | German→English | | | |
|---|---|---|---|---|---|---|---|---|
| | test-07 | test-08 | nt-14 | unk | test-07 | test-08 | nt-14 | unk |
| backtrans-bad | 18.86 | 19.27 | 20.49 | 3.22% | 14.66 | 14.62 | 15.07 | 1.45% |
| backtrans-good | 29.71 | 29.51 | 32.10 | 0.24% | 24.19 | 24.19 | 25.75 | 0.73% |
| backtrans-nmt | 31.10 | 31.43 | 31.27 | 0.0% | 26.02 | 26.03 | 26.98 | 0.0% |

Table 3: BLEU scores for (backward) translation into English

three MT systems of increasing quality:

1. `backtrans-bad`: this is a very poor SMT system trained using only 50k parallel sentences from the out-of-domain data, and no additional monolingual data. For this system as for the next one, we use Moses (Koehn et al., 2007) out-of-the-box, computing alignments with Fastalign (Dyer et al., 2013), with a minimal pre-processing (basic tokenization). This setting provides us with a pessimistic estimate of what we could get in low-resource conditions.

2. `backtrans-good`: these are much larger SMT systems, which use the same parallel data as the baseline NMTs (see § 2.2) and all the English monolingual data available for the WMT 2017 shared tasks, totalling approximately 174M sentences. These systems are strong, yet relatively cheap to build.

3. `backtrans-nmt`: these are the best NMT systems we could train, using settings that replicate the forward translation NMTs.

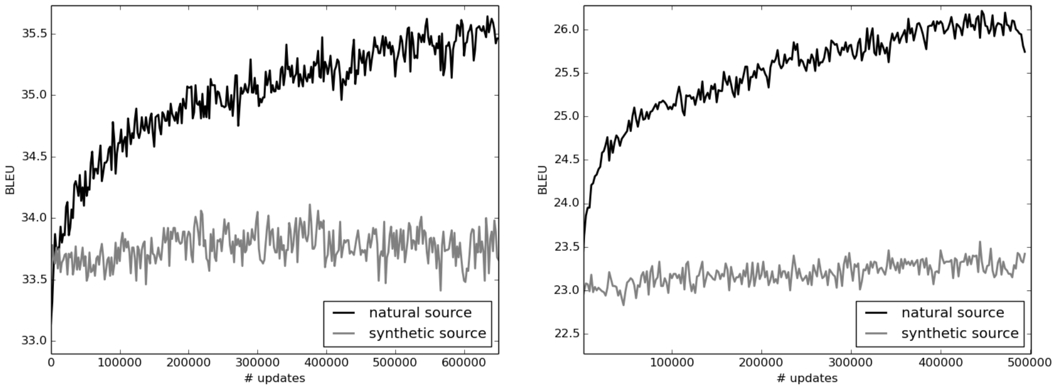
Note that we do not use any in-domain (*Europarl*) data to train these systems. Their performance is reported in Table 3, where we observe a 12 BLEU points gap between the worst and best systems (for both languages).

As noted *eg.* in (Park et al., 2017; Crego and Senellart, 2016), artificial parallel data obtained through *forward-translation* (FT) can also prove advantageous and we also consider a FT system (`fwdtrans-nmt`): in this case the *target* side of the corpus is artificial and is generated using the baseline NMT applied to a natural source.

### 3.1.2 BT quality does matter

Our results (see Table 2) replicate the findings of (Sennrich et al., 2016a): large gains can be obtained from BT (nearly +2 BLEU in French and German); better artificial data yields better translation systems. Interestingly, our best Moses system is almost as good as the NMT and an order of magnitude faster to train. Improvements obtained with the bad system are much smaller; contrary to the better MTs, this system is even detrimental for the out-of-domain test.

Gains with forward translation are significant, as in (Chinea-Rios et al., 2017), albeit about half as good as with BT, and result in small improvements for the in-domain and for the out-of-domain tests. Experiments combining forward and backward translation (`backfwdtrans-nmt`), each



English→French English→German

Figure 1: Learning curves from `backtrans-nmt` and `natural`. Artificial parallel data is more prone to overfitting than natural data.

using a half of the available artificial data, do not outperform the best BT results.

We finally note the large remaining difference between BT data and natural data, even though they only differ in their source side. This shows that at least in our domain-adaptation settings, BT does not really act as a regularizer, contrarily to the findings of (Poncelas et al., 2018; Sennrich et al., 2016b). Figure 3.1.1 displays the learning curves of these two systems. We observe that `backtrans-nmt` improves quickly in the earliest updates and then stays horizontal, whereas `natural` continues improving, even after 400k updates. Therefore BT does not help to avoid overfitting, it actually encourages it, which may be due "easier" training examples (cf. § 3.2).

### 3.2 Properties of back-translated data

Comparing the natural and artificial sources of our parallel data *wrt.* several linguistic and distributional properties, we observe that (see Fig. 2 - 3):

(i) artificial sources are on average shorter than natural ones: when using BT, cases where the source is shorter than the target are rarer; cases when they have the same length are more frequent.

(ii) automatic word alignments between artificial sources tend to be more monotonic than when using natural sources, as measured by the average Kendall $\tau$ of source-target alignments (Birch and Osborne, 2010): for French-English the respective numbers are 0.048 (natural) and 0.018 (artificial); for German-English 0.068 and 0.053. Using more monotonic sentence pairs turns out to be a facilitating factor for NMT, as also noted by Crego and Senellart (2016).

(iii) syntactically, artificial sources are simpler than real data; We observe significant differences in the distributions of tree depths.[3]

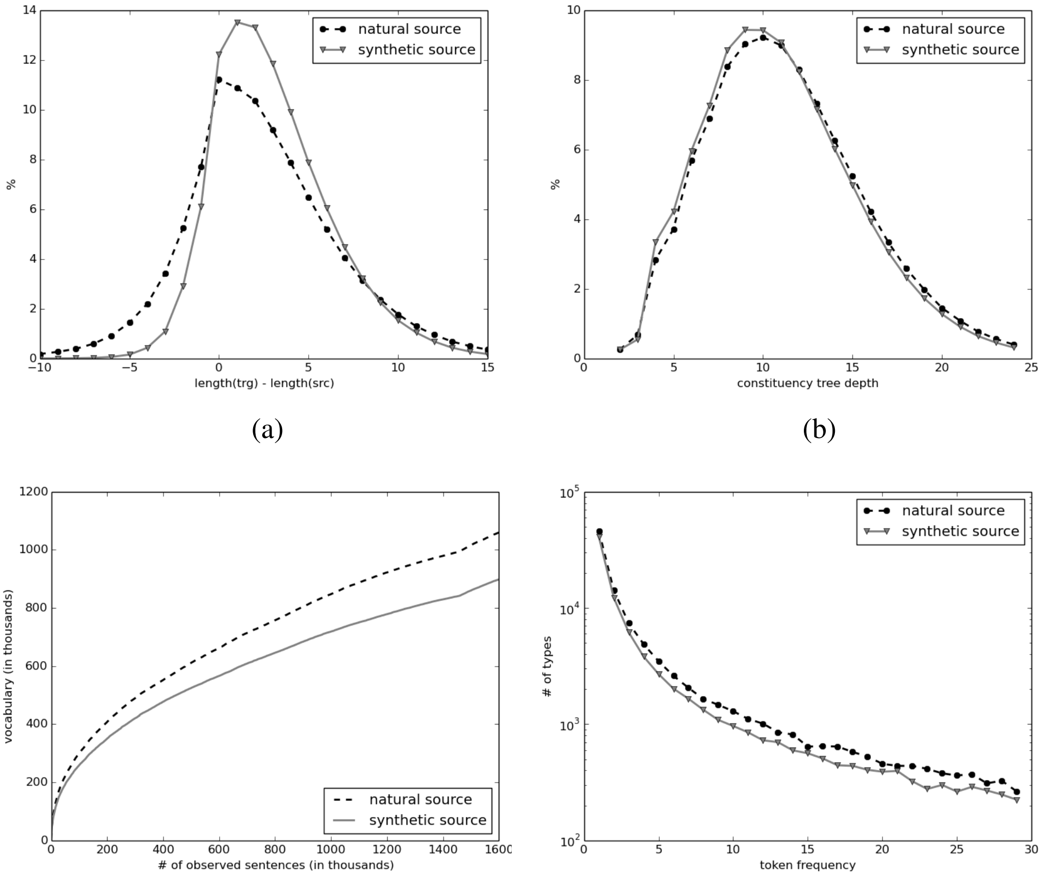
(iv) distributionally, plain word occurrences in artificial sources are more concentrated; this also translates into both a slower increase of the number of types *wrt.* the number of sentences and a smaller number of rare events.

The intuition is that properties (i) and (ii) should help translation as compared to natural source, while property (iv) should be detrimental. We checked (ii) by building systems with only 10M words from the natural parallel data selecting these data either randomly or based on the regularity of their word alignments. Results in Table 4 show that the latter is much preferable for the overall performance. This might explain that the mostly monotonic BT from Moses are almost as good as the fluid BT from NMT and that both boost the baseline.

## 4 Stupid Back-Translation

We now analyze the effect of using much simpler data generation schemes, which do not require the availability of a backward translation engine.

[3] Parses were automatically computed with CoreNLP (Manning et al., 2014).



(a)

(b)

(c)

(d)

Figure 2: Properties of pseudo-English data obtained with `backtrans-nmt` from French. The synthetic source contains shorter sentences (a) and slightly simpler syntax (b). The vocabulary growth *wrt.* an increasing number of observed sentences (c) and the token-type correlation (d) suggest that the natural source is lexically richer.

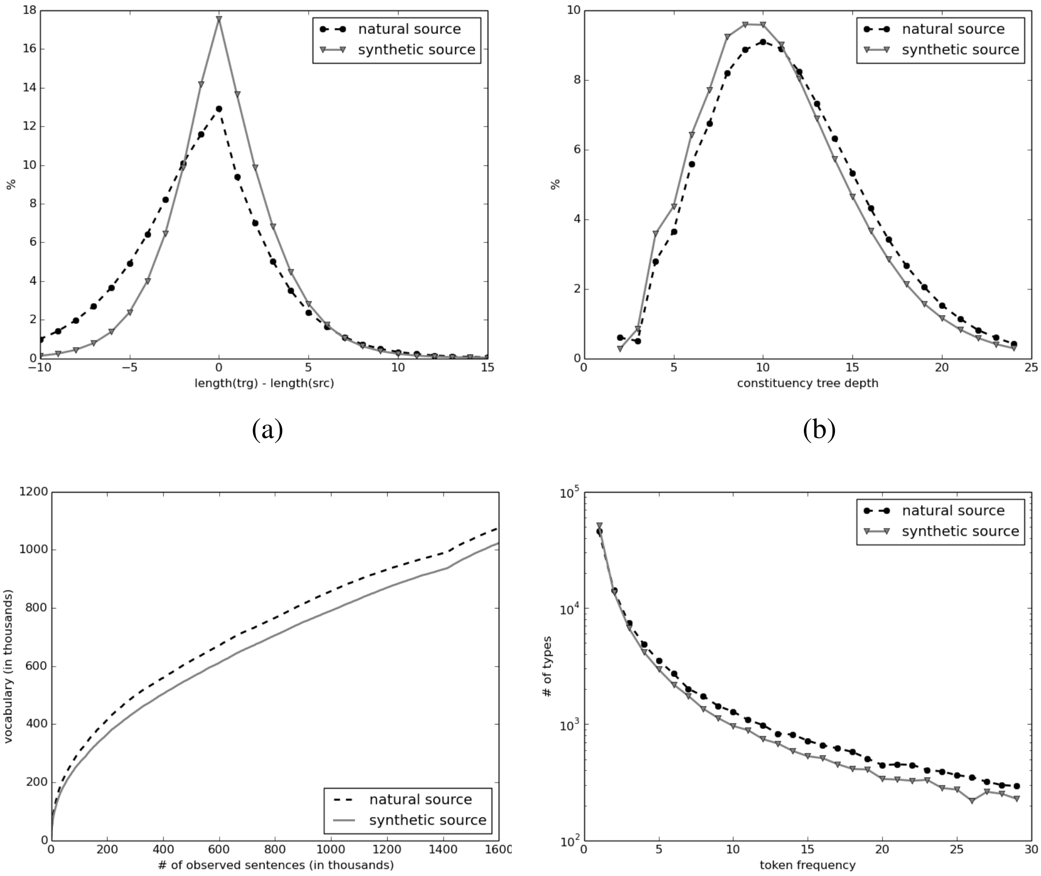
| | **test-07** | | | **test-08** | | | **newstest-14** | | |
|---|---|---|---|---|---|---|---|---|---|
| | BLEU | BEER | CTER | BLEU | BEER | CTER | BLEU | BEER | CTER |
| random | 32.08 | 62.98 | 50.78 | 32.66 | 62.86 | 49.99 | 23.05 | 55.38 | 58.51 |
| monotonic | 33.52 | 63.75 | 49.51 | 33.73 | 63.59 | 48.91 | 32.16 | 61.75 | 48.64 |

Table 4: Selection strategies for BT data (English-French)

## 4.1 Setups

We use the following cheap ways to generate pseudo-source texts:

1. `copy`: in this setting, the source side is a mere copy of the target-side data. Since the source vocabulary of the NMT is fixed, copying the target sentences can cause the occurrence of OOVs. To avoid this situation, Currey et al. (2017) decompose the target words into source-side units to make the copy look like source sentences. Each OOV found in the copy is split into smaller units until all the resulting chunks are in the source vocabulary.
2. `copy-marked`: another way to integrate copies without having to deal with OOVs is to augment the source vocabulary with a copy of the target vocabulary. In this setup, Ha et al. (2016) ensure that both vocabularies never overlap by marking the target word copies with a special language identifier. Therefore the English word *resume* cannot be confused with the homographic French word, which is marked *@fr@resume*.
3. `copy-dummies`: instead of using actual copies, we replace each word with “dummy” tokens. We use this unrealistic setup to observe the training over noisy and hardly informative source sentences.



(a)

(b)

(c)

(d)

Figure 3: Properties of pseudo-English data obtained with `backtrans-nmt` (back-translated from German). Tendencies similar to English-French can be observed and difference in syntax complexity is even more visible.

We then use the procedures described in § 2.2, except that the pseudo-source embeddings in the `copy-marked` setup are pretrained for three epochs on the in-domain data, while all remaining parameters are frozen. This prevents random parameters from hurting the already trained model.

### 4.2 Copy+marking+noise is not so stupid

We observe that the `copy` setup has only a small impact on the English-French system, for which the baseline is already strong. This is less true for English-German where simple copies yield a significant improvement. Performance drops for both language pairs in the `copy-dummies` setup.

We achieve our best gains with the `copy-marked` setup, which is the best way to use a copy of the target (although the performance on the out-of-domain tests is at most the same as the baseline). Such gains may look surprising, since the NMT model does not need to learn to translate but only to copy the source. This is indeed what happens: to confirm this, we built a fake test set having identical source and target side (in French). The average cross-entropy for this test set is 0.33, very close to 0, to be compared with an average cost of 58.52 when we process an actual source (in English). This means that the model has learned to copy words from source to target with no difficulty, even for sentences not seen in training. A follow-up question is whether training a copying task instead of a translation task limits the improvement: would the NMT learn better if the task was harder? To measure this, we introduce noise in the target sentences copied onto the source, following the procedure of Lample et al. (2017): it deletes random words and performs a small random permutation of the remaining words. Results (+ *Source noise*) show no difference for the French in-domain test sets, but bring the out-of-domain score to the level of the baseline. Finally, we observe a significant improvement on German in-domain

| **English→French** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **test-07** | | | **test-08** | | | **newstest-14** | | |
| | BLEU | BEER | CTER | BLEU | BEER | CTER | BLEU | BEER | CTER |
| Baseline | 31.25 | 62.14 | 51.89 | 32.17 | 62.35 | 50.79 | 33.06 | 61.97 | 48.56 |
| `copy` | 31.65 | 62.45 | 52.09 | 32.23 | 62.37 | 52.20 | 32.80 | 61.99 | 49.05 |
| `copy-dummies` | 30.89 | 62.06 | 52.07 | 31.51 | 61.98 | 51.46 | 31.43 | 60.92 | 50.58 |
| `copy-marked` | 32.01 | 62.66 | 51.57 | 32.31 | 62.52 | 51.46 | 32.33 | 61.55 | 49.44 |
| + Source noise | 31.87 | 62.52 | 52.69 | 32.64 | 62.55 | 51.63 | 33.04 | 62.11 | 48.47 |
| **English→German** | | | | | | | | | |
| | **test-07** | | | **test-08** | | | **newstest-14** | | |
| | BLEU | BEER | CTER | BLEU | BEER | CTER | BLEU | BEER | CTER |
| Baseline | 21.36 | 57.08 | 63.32 | 21.27 | 57.11 | 60.67 | 22.49 | 57.79 | 55.64 |
| `copy` | 22.15 | 57.95 | 61.49 | 21.95 | 57.72 | 59.58 | 22.59 | 57.83 | 55.44 |
| `copy-dummies` | 21.73 | 57.84 | 61.35 | 21.38 | 57.38 | 60.10 | 21.12 | 56.81 | 57.21 |
| `copy-marked` | 22.58 | 58.23 | 61.10 | 22.47 | 57.97 | 59.24 | 22.53 | 57.54 | 55.85 |
| + Source noise | 22.92 | 58.62 | 60.27 | 22.83 | 58.36 | 58.48 | 22.34 | 57.47 | 55.72 |

Table 5: Performance *wrt.* various stupid BTs

test sets, compared to the baseline (about +1.5 BLEU). This last setup is even almost as good as the `backtrans-nmt` condition (see § 3.1) for German. This shows that learning to reorder and predict missing words can more effectively serve our purposes than simply learning to copy.

## 5 Towards more natural pseudo-sources

Integrating monolingual data into NMT can be as easy as copying the target into the source, which already gives some improvement; adding noise makes things even better. We now study ways to make pseudo-sources look more like natural data, using the framework of Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), an idea borrowed from Lample et al. (2017)[4].

### 5.1 GAN setups

In our setups, we use a marked target copy, viewed as a *fake* source, which a *generator* encodes so as to fool a discriminator *trained* to distinguish a *fake* from a *natural* source. Our architecture contains two distinct encoders, one for the natural source and one for the pseudo-source. The latter acts as the generator ($G$) in the GAN framework, computing a representation of the pseudo-source that is then input to a discriminator ($D$), which has to sort natural from artificial encodings. $D$ assigns a probability of a sentence being natural.

During training, the cost of the discriminator is computed over two batches, one with natural (out-of-domain) sentences $\mathbf{x}$ and one with (in-domain) pseudo-sentences $\mathbf{x}'$. The discriminator is a bidirectional-Recurrent Neural Network (RNN) of dimension 1024. Averaged states are passed to a single feed-forward layer, to which a sigmoid is applied. It inputs encodings of natural ($E(\mathbf{x})$) and pseudo-sentences ($G(\mathbf{x}')$) and is trained to optimize:

$$\begin{aligned} J^{(D)} = &-\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{\text{real}}} \log D(E(\mathbf{x})) \\ &-\frac{1}{2}\mathbb{E}_{\mathbf{x}'\sim p_{\text{pseudo}}} \log(1 - D(G(\mathbf{x}'))) \end{aligned}$$

$G$'s parameters are updated to maximally fool $D$, thus the loss $J^{(G)}$:

$$J^{(G)} = -\mathbb{E}_{\mathbf{x}'\sim p_{\text{pseudo}}} \log D(G(\mathbf{x}'))$$

Finally, we keep the usual MT objective. ($\mathbf{s}$ is a real or pseudo-sentence):

$$J^{(\text{MT})} = \log p(\mathbf{y}|\mathbf{s}) = -\mathbb{E}_{\mathbf{s}\sim p_{\text{all}}} \log \text{MT}(\mathbf{s})$$

We thus need to train three sets of parameters: $\theta^{(D)}, \theta^{(G)}$ and $\theta^{(\text{MT})}$ (MT parameters), with $\theta^{(G)} \in \theta^{(\text{MT})}$. The pseudo-source encoder and embeddings are updated *wrt.* both $J^{(G)}$ and $J^{(\text{MT})}$. Following (Goyal et al., 2016), $\theta^{(G)}$ is updated only when $D$'s accuracy exceeds $75\%$. On the other hand, $\theta^{(D)}$ is not updated when its accuracy exceeds $99\%$. At each update, two batches are generated for each type of data, which are encoded with the real or pseudo-encoder. The encoder outputs serve to compute $J^{(D)}$ and $J^{(G)}$. Finally, the pseudo-source is encoded again (once $G$ is updated), both encoders are plugged into the translation model and the MT cost is backpropagated down to the real and pseudo-word embeddings. Pseudo-encoder and discriminator parameters are pre-trained for 10k updates. At test time, the pseudo-encoder is ignored and inference is run as usual.

[4] Our implementation is available at `https://github.com/franckbrl/nmt-pseudo-source-discriminator`

| **English→French** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **test-07** | | | **test-08** | | | **newstest-14** | | |
| | BLEU | BEER | CTER | BLEU | BEER | CTER | BLEU | BEER | CTER |
| Baseline | 31.25 | 62.14 | 51.89 | 32.17 | 62.35 | **50.79** | **33.06** | 61.97 | 48.56 |
| `copy-marked` | 32.01 | 62.66 | **51.57** | 32.31 | 62.52 | 51.46 | 32.33 | 61.55 | 49.44 |
| + GANs | 31.95 | 62.55 | 52.87 | 32.24 | 62.47 | 52.16 | 32.86 | 61.90 | 48.97 |
| `copy-marked` + noise | 31.87 | 62.52 | 52.69 | 32.64 | 62.55 | 51.63 | 33.04 | 62.11 | 48.47 |
| + GANs | **32.41** | **62.78** | 52.25 | **32.79** | **62.72** | 50.92 | 33.01 | **61.98** | **48.37** |
| `backtrans-nmt` | **33.30** | **63.33** | **50.02** | **33.39** | **63.09** | **49.48** | **34.11** | **62.76** | **46.94** |
| + Distinct encoders | 32.29 | 62.83 | 51.55 | 32.98 | 62.91 | 51.19 | 33.60 | 62.43 | 48.06 |
| + GANs | 32.91 | 63.08 | 51.17 | 33.24 | 62.93 | 50.82 | 33.77 | 62.42 | 47.80 |
| `natural` | 35.10 | 64.71 | 48.33 | 35.29 | 64.52 | 48.26 | 34.96 | 63.08 | 46.67 |
| **English→German** | | | | | | | | | |
| | **test-07** | | | **test-08** | | | **newstest-14** | | |
| | BLEU | BEER | CTER | BLEU | BEER | CTER | BLEU | BEER | CTER |
| Baseline | 21.36 | 57.08 | 63.32 | 21.27 | 57.11 | 60.67 | 22.49 | **57.79** | 55.64 |
| `copy-marked` | 22.58 | 58.23 | 61.10 | 22.47 | 57.97 | 59.24 | 22.53 | 57.54 | 55.85 |
| + GANs | 22.71 | 58.25 | 61.25 | 22.44 | 57.86 | 59.28 | **22.81** | 57.54 | 55.99 |
| `copy-marked` + noise | 22.92 | 58.62 | 60.27 | **22.83** | **58.36** | **58.48** | 22.34 | 57.47 | 55.72 |
| + GANs | **23.01** | **58.66** | **60.22** | 22.53 | 58.16 | 58.65 | 22.64 | 57.70 | **55.48** |
| `backtrans-nmt` | 23.00 | 59.12 | **58.31** | 23.10 | 58.85 | **56.67** | 22.91 | 58.12 | **54.67** |
| + Distinct encoders | 23.62 | 58.83 | 59.74 | 23.10 | **58.50** | 58.19 | 22.82 | **57.91** | 54.96 |
| + GANs | **23.65** | **58.85** | 59.70 | **23.20** | **58.50** | 58.22 | **23.00** | 57.89 | 55.15 |
| `natural` | 26.74 | 61.14 | 56.19 | 26.16 | 60.64 | 54.76 | 23.84 | 58.64 | 54.23 |

Table 6: Performance *wrt.* different GAN setups

| **English→French** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **test-07** | | | **test-08** | | | **newstest-14** | | |
| | BLEU | BEER | CTER | BLEU | BEER | CTER | BLEU | BEER | CTER |
| Baseline | 31.25 | 62.14 | **51.89** | 32.17 | 62.35 | **50.79** | 33.06 | 61.97 | 48.56 |
| `deep-fusion` | 31.85 | 62.52 | 52.27 | 32.25 | 62.40 | 51.64 | **33.65** | **62.40** | **48.24** |
| `copy-marked` + noise + GANs | **32.41** | **62.78** | 52.25 | **32.79** | **62.72** | 50.92 | 33.01 | 61.98 | 48.37 |
| `+deep-fusion` | 31.96 | 62.59 | 51.96 | 32.59 | 62.59 | 51.65 | 32.96 | 61.95 | 48.95 |
| **English→German** | | | | | | | | | |
| | **test-07** | | | **test-08** | | | **newstest-14** | | |
| | BLEU | BEER | CTER | BLEU | BEER | CTER | BLEU | BEER | CTER |
| Baseline | 21.36 | 57.08 | 63.32 | 21.27 | 57.11 | 60.67 | 22.49 | 57.79 | 55.64 |
| `deep-fusion` | 21.65 | 57.57 | 62.38 | 21.33 | 57.33 | 60.54 | **23.10** | **58.06** | **55.33** |
| `copy-marked` + noise + GANs | **23.01** | **58.66** | **60.22** | **22.53** | **58.16** | **58.65** | 22.64 | 57.70 | 55.48 |
| `+deep-fusion` | 23.07 | 58.50 | 60.47 | 22.86 | 58.18 | 58.76 | 22.64 | 57.46 | 55.85 |

Table 7: Deep-fusion model

### 5.2 GANs can help

Results are in Table 6, assuming the same fine-tuning procedure as above. On top of the `copy-marked` setup, our GANs do not provide any improvement in both language pairs, with the exception of a small improvement for English-French on the out-of-domain test, which we understand as a sign that the model is more robust to domain variations, just like when adding pseudo-source noise. When combined with noise, the French model yields the best performance we could obtain with stupid BT on the in-domain tests, at least in terms of BLEU and BEER. On the News domain, we remain close to the baseline level, with slight improvements in German.

A first observation is that this method brings stupid BT models closer to conventional BT, at a greatly reduced computational cost. While French still remains 0.4 to 1.0 BLEU below very good backtranslation, both approaches are in the same ballpark for German - may be because BTs are better for the former system than for the latter.

Finally note that the GAN architecture has two differences with basic `copy-marked`: (a) a distinct encoder for real and pseudo-sentence; (b) a different training regime for these encoders. To sort out the effects of (a) and (b), we reproduce the GAN setup with BT sentences, instead of copies. Using a separate encoder for the pseudo-source in the `backtrans-nmt` setup can be detrimental to performance (see Table 6): translation degrades in French for all metrics. Adding GANs on top of the pseudo-encoder was not able to make up for the degradation observed in French, but allowed the German system to slightly outperform backtrans-nmt. Even though this setup is unrealistic and overly costly, it shows that GANs are actually helping even good systems.

## 6 Using Target Language Models

In this section, we compare the previous methods with the use of a target side Language Model (LM). Several proposals exist in the literature to integrate LMs in NMT: for instance, Domhan and Hieber (2017) strengthen the decoder by integrating an extra, source independent, RNN layer in a conventional NMT architecture. Training is performed either with parallel, or monolingual data. In the latter case, word prediction only relies on the source independent part of the network.

### 6.1 LM Setup

We have followed Gulcehre et al. (2017) and reimplemented[5] their deep-fusion technique. It requires to first independently learn a RNN-LM on the in-domain target data with a cross-entropy objective; then to train the optimal combination of the translation and the language models by adding the hidden state of the RNN-LM as an additional input to the softmax layer of the decoder.

Our RNN-LMs are trained using dl4mt[6] with the target side of the parallel data and the Europarl corpus (about 6M sentences for both French and German), using a one-layer GRU with the same dimension as the MT decoder (1024).

### 6.2 LM Results

Results are in Table 7. They show that deep-fusion hardly improves the Europarl results, while we obtain about +0.6 BLEU over the baseline on newstest-2014 for both languages. deep-fusion differs from stupid BT in that the model is not directly optimized on the in-domain data, but uses the LM trained on Europarl to maximize the likelihood of the out-of-domain training data. Therefore, no specific improvement is to be expected in terms of domain adaptation, and the performance increases in the more general domain. Combining deep-fusion and copy-marked + noise + GANs brings slight improvements on the German in-domain test sets, and performance out of the domain remains near the baseline level.

## 7 Re-analyzing the effects of BT

As a follow up of previous discussions, we analyze the effect of BT on the internals of the network. Arguably, using a copy of the target sentence instead of a natural source should not be helpful for the encoder, but is it also the case with a strong BT? What are the effects on the attention model?

### 7.1 Parameter freezing protocol

To investigate these questions, we run the same fine-tuning using the copy-marked, backtrans-nmt and backtrans-nmt setups. Note that except for the last one, all training scenarios have access to same target training data. We intend to see whether the overall performance of the NMT system degrades when we selectively freeze certain sets of parameters, meaning that they are not updated during fine-tuning.

### 7.2 Results

BLEU scores are in Table 8. The backtrans-nmt setup is hardly impacted by selective updates: updating the only decoder leads to a degradation of at most 0.2 BLEU. For copy-marked, we were not able to freeze the source embeddings, since these are initialized when fine-tuning begins and therefore need to be trained. We observe that freezing the encoder and/or the attention parameters has no impact on the English-German system, whereas it slightly degrades the English-French one. This suggests that using artificial sources, even of the poorest quality, has a positive impact on all the components of the network, which makes another big difference with the LM integration scenario.

The largest degradation is for natural, where the model is prevented from learning from informative source sentences, which leads to a decrease of 0.4 to over 1.0 BLEU. We assume from these experiments that BT impacts most of all the decoder, and learning to encode a pseudo-source, be it a copy or an actual back-translation, only marginally helps to significantly improve the quality. Finally, in the fwdtrans-nmt setup, freezing the decoder does not seem to harm learning with a natural source.

[5] Our implementation is part of the Nematus toolkit (theano branch): https://github.com/EdinburghNLP/nematus/blob/theano/doc/deep_fusion_lm.md

[6] https://github.com/nyu-dl/dl4mt-tutorial

| | English→French | | | English→German | | |
|---|---|---|---|---|---|---|
| | **test-07** | **test-08** | **nt-14** | **test-07** | **test-08** | **nt-14** |
| Baseline | 31.25 | 32.17 | 33.06 | 21.36 | 21.27 | 22.49 |
| `backtrans-nmt` | 33.30 | 33.39 | 34.11 | 23.00 | 23.10 | 22.91 |
| + Freeze source embedd. | 33.20 | 33.24 | 34.16 | 22.84 | 22.85 | 23.00 |
| + Freeze encoder | 33.17 | 33.25 | 33.73 | 22.72 | 22.74 | 22.64 |
| + Freeze attention | 33.13 | 33.22 | 33.47 | 23.03 | 23.01 | 22.85 |
| `copy-marked` | 32.01 | 32.31 | 32.33 | 22.58 | 22.47 | 22.53 |
| + Freeze encoder | 31.70 | 32.39 | 32.90 | 22.59 | 22.30 | 22.81 |
| + Freeze attention | 31.59 | 32.39 | 32.54 | 22.55 | 22.13 | 22.69 |
| `fwdtrans-nmt` | 31.93 | 32.62 | 33.56 | 21.97 | 21.89 | 22.52 |
| + Freeze decoder | 31.84 | 32.62 | 33.35 | 21.91 | 21.65 | 13.61 |
| `natural` | 35.10 | 35.29 | 34.96 | 26.74 | 26.16 | 23.84 |
| + Freeze encoder | 34.02 | 34.25 | 34.09 | 24.95 | 25.08 | 23.44 |
| + Freeze attention | 34.13 | 34.42 | 34.19 | 25.13 | 24.97 | 23.35 |

Table 8: BLEU scores with selective parameter freezing

## 8 Related work

The literature devoted to the use of monolingual data is large, and quickly expanding. We already alluded to several possible ways to use such data: using back- or forward-translation or using a target language model. The former approach is mostly documented in (Sennrich et al., 2016a), and recently analyzed in (Park et al., 2017), which focus on fully artificial settings as well as pivot-based artificial data; and (Poncelas et al., 2018), which studies the effects of increasing the size of BT data. The studies of Crego and Senellart (2016); Park et al. (2017) also consider forward translation and Chinea-Rios et al. (2017) expand these results to domain adaptation scenarios. Our results are complementary to these earlier studies.

As shown above, many alternatives to BT exist. The most obvious is to use target LMs (Domhan and Hieber, 2017; Gulcehre et al., 2017), as we have also done here; but attempts to improve the encoder using multi-task learning also exist (Zhang and Zong, 2016).

This investigation is also related to recent attempts to consider supplementary data with a valid target side, such as multi-lingual NMT (Firat et al., 2016), where source texts in several languages are fed in the same encoder-decoder architecture, with partial sharing of the layers. This is another realistic scenario where additional resources can be used to selectively improve parts of the model.

Round trip training is another important source of inspiration, as it can be viewed as a way to use BT to perform semi-unsupervised (Cheng et al., 2016) or unsupervised (He et al., 2016) training of NMT. The most convincing attempt to date along these lines has been proposed by Lample et al. (2017), who propose to use GANs to mitigate the difference between artificial and natural data.

## 9 Conclusion

In this paper, we have analyzed various ways to integrate monolingual data in an NMT framework, focusing on their impact on quality and domain adaptation. While confirming the effectiveness of BT, our study also proposed significantly cheaper ways to improve the baseline performance, using a slightly modified copy of the target, instead of its full BT. When no high quality BT is available, using GANs to make the pseudo-source sentences closer to natural source sentences is an efficient solution for domain adaptation.

To recap our answers to our initial questions: the quality of BT actually matters for NMT (cf. § 3.1) and it seems that, even though artificial source are lexically less diverse and syntactically complex than real sentence, their monotonicity is a facilitating factor. We have studied cheaper alternatives and found out that copies of the target, if properly noised (§ 4), and even better, if used with GANs, could be almost as good as low quality BTs (§ 5): BT is only worth its cost when good BT can be generated. Finally, BT seems preferable to integrating external LM - at least in our data condition (§ 6). Further experiments with larger LMs are needed to confirm this observation, and also to evaluate the complementarity of both strategies. More work is needed to better understand the impact of BT on subparts of the network (§ 7).

In future work, we plan to investigate other cheap ways to generate artificial data. The experimental setup we proposed may also benefit from a refining of the data selection strategies to focus on the most useful monolingual sentences.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the first International Conference on Learning Representations*, San Diego, CA.

Alexandra Birch and Miles Osborne. 2010. LRscore for evaluating lexical and reordering quality in MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 327–332, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 330–336, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1965–1974. Association for Computational Linguistics.

Mara Chinea-Rios, Álvaro Peris, and Francisco Casacuberta. 2017. Adapting neural machine translation with parallel synthetic data. In *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers*, pages 138–147, Copenhagen, Denmark. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Josep Maria Crego and Jean Senellart. 2016. Neural machine translation from simplified translations. *CoRR*, abs/1612.06139.

Annad Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark. Association for Computational Linguistics.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505, Copenhagen, Denmark. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia.

Andreas Eisele and Yu Chen. 2010. MultiUN: A Multilingual Corpus from United Nation Documents. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 2868–2872. European Language Resources Association (ELRA).

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 4601–4609, Barcelona, Spain.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017. On integrating a language model into neural machine translation. *Comput. Speech Lang.*, 45(C):137–148.

Thanh-Le Ha, Jan Niehues, and Alex Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of the 13th International Workshop on Spoken Language Translation (IWSLT 2016)*, Seattle, WA, USA.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 820–828. Curran Associates, Inc.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Philipp Koehn. 2005. A parallel corpus for statistical machine translation. In *Proc. MT-Summit*, Phuket, Thailand.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical MT. In *Proc. ACL:Systems Demos*, pages 177–180, Prague, Czech Republic.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA.

Jaehong Park, Jongyoon Song, and Sungroh Yoon. 2017. Building a neural machine translation system using only synthetic parallel data. *CoRR*, abs/1704.00253.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating backtranslation in neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, EAMT, Alicante, Spain.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne. 2014. Billions of parallel words for free: Building and using the eu bookshop corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Miloš Stanojević and Khalil Sima'an. 2014. Fitting sentence level translation evaluation with many dense features. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 202–206, Doha, Qatar. Association for Computational Linguistics.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation Edit Rate on Character Level. In *Proceedings of the First Conference on Machine Translation*, pages 505–510, Berlin, Germany. Association for Computational Linguistics.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

# Neural Machine Translation into Language Varieties

**Surafel M. Lakew**[†⋆], **Aliia Erofeeva**[†], **Marcello Federico**[⋆+]
[†]University of Trento, [⋆]Fondazione Bruno Kessler, [+]MMT Srl, Trento, Italy
[†]name.surname@unitn.it, [⋆]surname@fbk.eu

## Abstract

Both research and commercial machine translation have so far neglected the importance of properly handling the spelling, lexical and grammar divergences occurring among language varieties. Notable cases are standard national varieties such as Brazilian and European Portuguese, and Canadian and European French, which popular online machine translation services are not keeping distinct. We show that an evident side effect of modeling such varieties as unique classes is the generation of inconsistent translations. In this work, we investigate the problem of training neural machine translation from English to specific pairs of language varieties, assuming both labeled and unlabeled parallel texts, and low-resource conditions. We report experiments from English to two pairs of dialects, European-Brazilian Portuguese and European-Canadian French, and two pairs of standardized varieties, Croatian-Serbian and Indonesian-Malay. We show significant BLEU score improvements over baseline systems when translation into similar languages is learned as a multilingual task with shared representations.

## 1 Introduction

The field of machine translation (MT) is making amazing progress, thanks to the advent of neural models and deep learning. While just few years ago research in MT was struggling to achieve *useful* translations for the most requested and high-resourced languages, the level of translation quality reached today has raised the demand and interest for less-resourced languages and the solution of more subtle and interesting translation tasks (Bentivogli et al., 2018). If the goal of machine translation is to help worldwide communication, then the time has come to also cope with dialects or more generally language varieties[1]. Remarkably, up to now, even standard national language varieties, such as Brazilian and European Portuguese, or Canadian and European French, which are used by relatively large populations have been quite neglected both by research and industry. Prominent online commercial MT services, such as Google Translate and Bing, are currently not offering any variety of Portuguese and French. Even worse, systems offering such languages tend to produce inconsistent outputs, like mixing lexical items from different Portuguese (see for instance the translations shown in Table 1). Clearly, in the perspective of delivering high-quality MT to professional post-editors and final users, this problem urges to be fixed.

While machine translation from many to one varieties is intuitively simpler to approach[2], it is the opposite direction that presents the most relevant problems. First, languages varieties such as dialects might significantly overlap thus making differences among their texts quite subtle (e.g., particular grammatical constructs or lexical divergences like the ones reported in the example). Second, parallel data are not always labeled at the level of language variety, making it hard to develop specific NMT engines. Finally, training data might be very unbalanced among different varieties, due to the population sizes of their respective speakers or for other reasons. This clearly makes it harder to model the lower-resourced varieties (Koehn and Knowles, 2017).

In this work we present our initial effort to systematically investigate ways to approach NMT from English into four pairs of language varieties:

[1] In sociolinguistics, a variety is a specific form of language, that may include dialects, registers, styles, and other forms of language, as well as a standard language. See Wardhaugh (2006) for a more comprehensive introduction.

[2] We will focus on this problem in future work and disregard possible varieties in the source side, such as American and British English, in this work.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 156–164
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64016

| English (source) | I'm going to the gym before breakfast. No, I'm not going to the gym. |
|---|---|
| pt (Google Translate) | Eu estou indo para a academia antes do café da manhã. Não, eu não vou ao ginásio. |
| pt-BR (M-C2) | Eu vou á academia antes do café da manhã. Não, eu não vou à academia. |
| pt-EU (M-C2) | Vou para o ginásio antes do pequeno-almoço. Não, não vou para o ginàsio. |
| pt-BR (M-C2_L) | Vou à academia antes do café da manhã. Não, não vou à academia. |
| pt-PT (M-C2_L) | Vou ao ginásio antes do pequeno-almoço. Não, não vou ao ginásio. |

Table 1: MT from English into Portuguese varieties. Example of mixed translations generated by Google Translate (as of 20th July, 2018) and translations generated by our variety-specific models. For the underlined English terms both their Brazilian and European translation variants are shown.

Portuguese European - Portuguese Brazilian, European French - Canadian French, Serbian - Croatian, and Indonesian - Malay[3]. For each couple of varieties, we assume to have both parallel text labeled with the corresponding couple member, and parallel text without such information. Moreover, the considered target pairs, while all being mutually intelligible, present different levels of linguistic similarity and also different proportions of available training data. For our tasks we rely on the WIT[3] TED Talks collection[4], used for the International Workshop of Spoken Language Translation, and OpenSubtitles2018, a corpus of subtitles available from the OPUS collection[5].

After presenting related work (Section 2) on NLP and MT of dialects and related languages, we introduce (in Section 3) baseline NMT systems, either language/dialect specific or generic, and multilingual NMT systems, either trained with fully supervised (or labeled) data or with partially supervised data. In Section 4, we introduce our datasets, NMT set-ups based on the Transformer architecture, and then present the results for each evaluated system. We conclude the paper with a discussion and conclusion in Sections 5 and 6.

## 2 Related work

### 2.1 Machine Translation of Varieties

Most of the works on translation between and from/to written language varieties involve rule-based transformations, e.g., for European and Brazilian Portuguese (Marujo et al., 2011), Indonesian and Malay (Tan et al., 2012), Turkish and Crimean Tatar (Altintas and Çiçekli, 2002); or phrase-based statistical MT (SMT) systems, e.g., for Croatian, Serbian, and Slovenian (Popović et al., 2016), Hindi and Urdu (Durrani et al., 2010), or Arabic dialects (Harrat et al., 2017). Notably, Pourdamghani and Knight (2017) build an unsupervised deciphering model to translate between closely related languages without parallel data. Salloum et al. (2014) handle mixed Arabic dialect input in MT by using a sentence-level classifier to select the most suitable model from an ensemble of multiple SMT systems. In NMT, however, there have been fewer studies addressing language varieties. It is reported that an RNN model outperforms SMT when translating from Catalan to Spanish (Costa-jussà, 2017) and from European to Brazilian Portuguese (Costa-Jussà et al., 2018). Hassan et al. (2017) propose a technique to augment training data for under-resourced dialects via projecting word embeddings from a resource-rich related language, thus enabling training of dialect-specific NMT systems. The authors generate spoken Levantine-English data from larger Arabic-English corpora and report improvement in BLEU scores compared to a low-resourced NMT model.

### 2.2 Dialect Identification

A large body of research in dialect identification stems from the DSL shared tasks (Zampieri et al., 2014, 2015; Malmasi et al., 2016; Zampieri et al., 2017). Currently, the best-performing methods include linear machine learning algorithms such as SVM, naïve Bayes, or logistic regression, which use character and word $n$-grams as features and are usually combined into ensembles (Jauhiainen et al., 2018). Tiedemann and Ljubeši (2012) present the idea of leveraging parallel corpora for language identification: content comparability allows capturing subtle linguistic differences between dialects while avoiding content-related biases. The problem of ambiguous sentences, i.e., those for which it is impossible to decide upon the dialect tag, has been demonstrated for Portuguese by Goutte et al. (2016) through inspection of disagreement between human annotators.

[3] According to Wikipedia, Brazilian Portuguese is a dialect of European Portuguese, Canadian French is a dialect of European French, Serbian and Croatian are standardized registers of Serbo-Croatian, and Indonesian is a standardized register of Malay.

[4] http://wit3.fbk.eu/

[5] http://opus.nlpl.eu/

### 2.3 Multilingual NMT

In a *one-to-many* multilingual translation scenario, Dong et al. (2015) proposed a multi-task learning approach that utilizes a single encoder for source languages and separate attention mechanisms and decoders for every target language. Luong et al. (2015) used distinct encoder and decoder networks for modeling language pairs in a *many-to-many* setting. Firat et al. (2016) introduced a way to share the attention mechanism across multiple languages. A simplified and efficient multilingual NMT approach is proposed by Johnson et al. (2016) and Ha et al. (2016) by prepending language tokens to the input string. This approach has greatly simplified multi-lingual NMT, by eliminating the need of having separate encoder/decoder networks and attention mechanism for every new language pair. In this work we follow a similar strategy, by incorporating an artificial token as a unique *variety flag*.

## 3 NMT into Language Varieties

Our assumption is to translate from language $E$ (English) into each of two varieties $A$ and $B$. We assume to have parallel training data $D_{E\to A}$ and $D_{E\to B}$ for each variety as well as unlabeled data $D_{E\to A\cup B}$. For the sake of experimentation we consider three application scenarios in which a fixed amount of parallel training data $E$-$A$ and $E$-$B$ is partitioned in different ways:

- *Supervised*: all sentence pairs are respectively put in $D_{E\to A}$ and $D_{E\to B}$, leaving $D_{E\to A\cup B}$ empty;

- *Unsupervised*: all sentence pairs are jointly put in $D_{E\to A\cup B}$, leaving $D_{E\to A}$ and $D_{E\to B}$ empty;

- *Semi-supervised*: two-third of $E$-$A$ and $E$-$B$ are, respectively, put in $D_{E\to A}$ and $D_{E\to B}$, and the remaining sentence pairs are put in $D_{E\to A\cup B}$.

**Supervised and Unsupervised Baselines.** For each translation direction we compare three baseline NMT systems. The first system is an unsupervised generic (`Gen`) system trained on the union of the language varieties training data. Notice that `Gen` makes no distinction between $A$ and $B$ and uses all data in an unsupervised way. The second is a supervised variety-specific system (`Spec`) trained on the corresponding language variety training set. The third system (`Ada`) is obtained by adapting the `Gen` system to a specific variety.[6] Adaptation is carried out by simply restarting the training process from the generic model using all the available variety specific training data.

**Supervised Multilingual NMT.** We build on the idea of multilingual NMT (`Mul`), where one single NMT system is trained on the union of $D_{E\to A}$ and $D_{E\to B}$. Each source sentence both at training and inference time is prepended with the corresponding target language variety label ($A$ or $B$). Notice that the multilingual architecture leverages the target forcing symbol both as input to the encoder to build its states, and as initial input to the decoder to trigger the first target word.

**Semi-Supervised Multilingual NMT.** We consider here multilingual NMT models that make also use of unlabeled data $D_{E\to A\cup B}$. The first model we propose, named `M-U`, uses the available data $D_{E\to A}$, $D_{E\to B}$ and $D_{E\to A\cup B}$ as they are, by not specifying any label at training time for entries from $D_{E\to A\cup B}$. The second model, named `M-C2`, works similarly to `Mul`, but relying on a language variety identification module (trained on the target data of $D_{E\to A}$ and $D_{E\to B}$) that maps each unlabeled data point either to $A$ or $B$. The third model, named `M-C3`, can be seen as an enhancement of `M-U`, as the unlabeled data is automatically classified into one of three classes: $A$, $B$, or $A\cup B$. For the third class, like with `M-U`, no label is applied on the source sentence.

## 4 Experimental Set-up

### 4.1 Dataset and Preprocessing

The experimental setting consists of eight target varieties and English as source. We use publicly available datasets from the WIT$^3$ TED corpus (Cettolo et al., 2012). The summary of the partitioned training, dev, and test sets are given in Table 2, where Tr. 2/3 is the labeled portion of the training set used to train the semi-supervised models, while the other 1/3 are either held out as unlabeled (`M-U`) or classified automatically (`M-C2`, `M-C3`). In the preprocessing stages, we tokenize the corpora and remove lines longer than 70 tokens. The Serbian corpus written in Cyrillic is transliterated into Latin script with CyrTranslit[7]. In addition, to also run a large-data experiment,

[6] We test this system only on the Portuguese varieties.

[7] https://pypi.org/project/cyrtranslit

| | Train | Ratio (%) | Tr. 2/3 | Dev | Test |
|---|---|---|---|---|---|
| pt-BR | 234K | 58.23 | 156K | 1567 | 1454 |
| pt-EU | 168K | 47.77 | 56K | 1565 | 1124 |
| fr-CA | 18K | 10.26 | 12K | 1608 | 1012 |
| fr-EU | 160K | 89.74 | 106K | 1567 | 1362 |
| hr | 110K | 54.20 | 73K | 1745 | 1222 |
| sr | 93K | 45.80 | 62K | 1725 | 1214 |
| id | 105k | 96.71 | 70K | 932 | 1448 |
| ms | 3.6K | 3.29 | 2.4k | 1024 | 738 |
| pt-BR_L | 47.2M | 64.91 | 31.4M | 1567 | 1454 |
| pt-EU_L | 25.5M | 35.10 | 17M | 1565 | 1124 |

Table 2: Number of parallel sentences of the TED Talks used for training, development and testing. At the bottom, the large-data set-up which uses the OpenSubtitles (pt-BR_L and pt-PT_L) as additional training set.

we expand the English−European/Brazilian Portuguese data with the corresponding OpenSubtitles2018 datasets from the OPUS corpus. Table 2 summarizes the augmented training data, while keeping the same dev and test sets.

### 4.2 Experimental Settings

We trained all systems using the Transformer model[8] (Vaswani et al., 2018). We use the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.2 and a dropout also set to 0.2. A shared source and target vocabulary of size 16k is generated via sub-word segmentation (Wu et al., 2016). The choice for the vocabulary size follows the recommendations in Denkowski and Neubig (2017) regarding training of NMT systems on TED Talks data. Overall we use a uniform setting for all our models, with a 512 embedding dimension and hidden units, and 6 layers of self-attention encoder-decoder network. The training batch size is of 6144 sub-word tokens and the max length after segmentation is set to 70. Following Vaswani et al. (2017) and for a fair comparison, experiments are run for 100k training steps, i.e., in the low-resource settings all models are observed to converge within these steps. Adaptation experiments are run to convergence, which requires roughly half of the steps (i.e., 50k) required to train the generic low-resource model. On the other hand, large-data systems are trained for up to 800k steps, which also showed to be a convergence point. For the final evaluation we take the best performing checkpoint on the dev set. All models are trained using Tesla V100-pcie-16gb on a single GPU.

| | pt | sr-hr | fr | id-ms | pt_L |
|---|---|---|---|---|---|
| ROC AUC | 82.29 | 88.12 | 80.99 | 81.99 | 52.75 |

Table 3: Performance of language identification on the low-resource and high-resource (pt_L) settings

### 4.3 Language Variety Identification

To automatically identify the language variety of unlabeled target sentences, we train a fastText model (Joulin et al., 2017), a simple yet efficient linear bag of words classifier. We use both word- and character-level $n$-grams as features. In the low-resource condition, we train the classifier on the 2/3 portion of the labeled training data. For the large-data experiment, instead, we used a relatively smaller and independent corpus consisting of 3.3 million pt-BR−pt-EU parallel sentences extracted from OpenSubtitles2018 after filtering out identical sentences pairs and sentences occurring (in any of the two varieties) in the NMT training data. Additionally, low-resource training sentences (fr-CA and ms) are randomly oversampled to mitigate class imbalance.

For each pair of varieties, we train five base classifiers differing in random initialization. In the M-C2 experiments, prediction is determined based on soft fusion voting, i.e., the final label is the argmax of the sum of class probabilities. Due to class skewness in the evaluation set, we report binary classification performance in terms of ROC AUC (Fawcett, 2006) instead of accuracy in Table 3. For M-C3 models, we handle ambiguous examples using the majority voting scheme: in order for a label to be assigned, its softmax probability should be strictly higher than fifty percents according to the majority of the base classifiers, otherwise no tag is applied. On average, this resulted in <1% of unlabeled sentences for the small data condition, and about 2% of unlabeled sentences for the large data condition.

## 5 Results and Discussion

We run experiments with all the systems introduced in Section 3, on four pairs of languages varieties. Results are reported in Table 4 for the low-resource setting and in Table 5 for the large data setting.

### 5.1 Low-resource setting

Among the supervised models, which are using all the available training data, the multilingual NMT model Mul outperforms the variety-specific

[8] https://github.com/tensorflow/tensor2tensor

|  |  | pt-BR | pt-EU | average |
|---|---|---|---|---|
| Unsuper. | Gen | ↓36.52 | ↓33.75 | 35.14 |
| Supervis. | Spec | ↓35.85 | ↓35.84 | 35.85 |
| ” | Ada | ↓36.54 | ↓36.59 | 36.57 |
| ” | Mul | **37.86** | **38.42** | **38.14** |
| Semi-sup. | M-U | ↓37.09 | 37.59 | 37.34 |
| ” | M-C2 | **37.70** | **38.35** | **38.03** |
| ” | M-C3 | 37.59 | 38.31 | 37.95 |

|  |  | fr-EU | fr-CA | average |
|---|---|---|---|---|
| Unsuper. | Gen | **33.91** | ↓30.91 | 32.41 |
| Supervis. | Spec | 33.52 | ↓17.13 | 25.33 |
| ” | Mul | 33.40 | **37.37** | **35.39** |
| Semi-sup. | M-U | 33.28 | 37.96 | 35.62 |
| ” | M-C2 | 33.79 | ↑38.60 | 36.20 |
| ” | M-C3 | ↑**34.16** | ↑**39.30** | **36.73** |

|  |  | hr | sr | average |
|---|---|---|---|---|
| Unsuper. | Gen | ↓21.71 | ↓19.20 | 20.46 |
| Supervis. | Spec | ↓22.50 | ↓19.92 | 21.21 |
| ” | Mul | **23.99** | **21.37** | **22.68** |
| Semi-sup. | M-U | **24.30** | 21.53 | 22.91 |
| ” | M-C2 | 24.14 | 21.26 | 22.70 |
| ” | M-C3 | 24.22 | **21.97** | **23.10** |

|  |  | id | ms | average |
|---|---|---|---|---|
| Unsuper. | Gen | 26.56 | ↓13.86 | 20.21 |
| Supervis. | Spec | 26.20 | ↓2.73 | 14.47 |
| ” | Mul | **26.66** | **15.77** | **21.22** |
| Semi-sup. | M-U | **26.52** | 15.58 | 21.05 |
| ” | M-C2 | 26.36 | **16.31** | **21.34** |
| ” | M-C3 | 26.40 | 15.23 | 20.82 |

Table 4: BLEU scores of the presented models, trained with unsupervised, supervised and semi-supervised data, from English to Brazilian Portuguese (pt-BR) and European Portuguese (pt-EU), Canadian French (fr-CA) and European French (fr-EU), Croatian (hr) and Serbian (sr), and Indonesian (id) and Malay (ms). Arrows ↓↑ indicate statistically significant differences calculated against `Mul` using bootstrap resampling with $\alpha = 0.05$ (Koehn, 2004).

models on all considered directions. Remarkably, the `Mul` model also outperforms the adapted `Ada` model on the available translation directions. The unsupervised generic model `Gen`, that mixes together all the available data, as expected tends to perform better than the supervised specific models of the less resourced varieties. Particularly, this improvement is observed for Malay (ms) and Canadian French (fr-CA), which respectively represent the 3.3% and 10% of the overall training data used by their corresponding (`Gen`) systems. On the contrary, a degradation is observed for European Portuguese (pt-Eu) and Serbian (sr), which represent 42% and 45% of their respective training sets. Even though very low-resourced varieties can benefit from the mix, it is also evident that the `Gen` model can easily get biased because of the imbalance between the datasets.

In the semi-supervised scenario, we report results with three multilingual systems that integrate the 1/3 of unlabeled data to the training corpus in three different ways: *(i)* without labels (`M-U`), *(ii)* with automatic labels forcing one of two possible classes (`M-C2`), *(iii)* with automatic labels of one of the two options or no label in case of low confidence of the classifier (`M-C3`).

Results show that on average automatic tagging of the unlabeled data is better than leaving them unlabeled, although `M-U` still remains a better choice than using specialized and generic systems. The best between `M-C2` and `M-C3` performs on average from very close to better than the best supervised method.

If we look at the single language variety, the obtained figures are not showing a coherent picture. In particular, in the Croatian-Serbian and Indonesian-Malay pairs the best resourced language seems to benefit more from keeping the data unlabeled (`M-U`). Interestingly, even the worst semi-supervised model performs very close or even better than the best supervised model, which suggests the importance of taking advantage of all available data even if they are not labeled.

Focusing on the statistically significant improvements, the best supervised (`Mul`) is better than the unsupervised (`Gen`), whereas the best semi-supervised (`M-C2` or `M-C3`) is either comparable or better than the best supervised.

### 5.2 High-resource setting

Unlike what observed in the low-resource setting, where `Mul` outperforms `Spec` in the supervised scenario, in the large data condition, variety specific models apparently seem the best choice. Notice, however, that the supervised multilingual system `Mul` provides just a slightly lower level of performance with a simpler architecture (one network in place of two). The unsupervised generic model `Gen`, trained with the mix of the two varieties datasets, performs significantly worse than the other two supervised approaches, this is particularly visible for the pt-EU direction. Very

|  |  | pt-BR | pt-EU | average |
|---|---|---|---|---|
| Unsuper. | Gen | ↓ 39.78 | ↓ 36.13 | 37.96 |
| Supervis. | Spec | **41.54** | **40.42** | **40.98** |
| ” | Mul | 41.28 | 40.28 | 40.78 |
| Semi-sup. | M-U | 41.21 | 39.88 | 40.55 |
| ” | M-C2 | 41.20 | 40.02 | 40.61 |
| ” | M-C3 | **41.56** | **40.22** | **40.89** |

Table 5: BLEU score on the test set of models trained with large-scale data, from English to Brazilian Portuguese (pt-BR) and European Portuguese (pt-EU). Arrows ↓↑ indicate statistically significant differences calculated against the Mul model.

likely, in addition to the ambiguities that arise from naively mixing the data of the two different dialects, there is also a bias effect towards pt-BR which is due to the very unbalanced proportions of data between the two dialects (almost 1:2).

Hence, in the considered high-resource setting, the Spec and Mul models result as best possible solutions against which comparing our semi-supervised approaches.

In the semi-supervised scenario, the obtained results confirm that our approach of automatically classifying the unlabeled data $D_{E \rightarrow A \cup B}$ improves over using the data as they are (M-U). Nevertheless, M-U still confirms to perform better than the fully unlabeled Gen model. In both translation directions, M-C2 and M-C3 get quite close to the performance of the supervised Spec model. In particular, M-C3 shows to outperform the M-C2 model, and even outperforms on average the supervised Mul model. In other words, the semi-supervised model leveraging three-class automatic labels (of $D_{E \rightarrow A \cup B}$) seems to perform better than the supervised model with two dialect labels. Besides the comparable BLEU scores, the supervised (Spec and Mul) perform in statistically insignificant way against the best semi-supervised (M-C3), although outperforming the unsupervised (Gen) model.

This result raises the question if relabeling all the training data can be a better option than using a combination of manual and automatic labels. This issue is investigated in the next subsection.

### Unsupervised Multilingual Models

As discussed in Section 4.3, the language classifier for the large-data condition is trained on dialect-to-dialect parallel data that does not overlap with the NMT training data. This condition permits hence to investigate a fully unsupervised training condition. In particular, we assume that all the available training data is unlabeled and create automatic language labels for all 47.2M sentences of pt-BR and 25.5M sentences of pt-EU (see Table 2). In a similar way as in Table 5, we keep the experimental setting of M-C2 and M-C3 models.

|  |  | pt-BR | pt-EU | average |
|---|---|---|---|---|
| Unsuper. | M-C2 | 41.50 | **40.21** | 40.86 |
| ” | M-C3 | **41.66** | 40.13 | **40.90** |

Table 6: BLEU scores on the test set by large scale multi-lingual models trained under an unsupervised condition, where all the training data are labeled automatically.

Table 6 reports the results of the multilingual models trained under the above described unsupervised condition. In comparison with the semi-supervised condition, both M-C2 and M-C3 show a slight performance improvement. In particular, the three-label M-C3 performs on average slightly better than the two-label M-C2 model. Actually, the little difference is justified by the fact that the classifier used the “third” label only for 6% of the data. Remarkably, despite the relatively low performance of the classifier, average score of the best unsupervised model M-C2 is almost on par with the supervised model Mul.

### 5.3 Translation Examples

Finally, in Table 7, we show an additional translation example produced by our semi-supervised multilingual models (both under low and high resource conditions) translating into the Portuguese varieties. For comparison we also include output from Google Translate which offers only a generic English-Portuguese direction. In particular, the examples contain the word *refrigerator* that has specific dialect variants. All our variety-specific systems show to generate consistent translations of this term, while Google Translate prefers to use the Brazilian translation variants for these sentences.

## 6 Conclusions

We presented initial work on neural machine translation from English into dialects and related languages. We discussed both situations where parallel data is supplied or not supplied with target language/dialect labels. We introduced and compared different neural MT models that can be

| | |
|---|---|
| English (source) | We offer a considerable number of different refrigerator models. We have also developed a new type of refrigerator. These include American-style side-by-side refrigerators. |
| pt (Google Translate) | ferecemos um número considerável de modelos diferentes de refrigeradores. Nós também desenvolvemos um novo tipo de geladeira. Estes incluem refrigeradores lado a lado estilo americano. |
| **Low-resource models** | |
| pt-BR (M-C2) | Nós oferecemos um número considerável de diferentes modelos de refrigerador. Também desenvolvemos um novo tipo de refrigerador. Eles incluem o estilo americano nas geladeiras lado a lado. |
| pt-EU (M-C2) | Oferecemos um número considerável de modelos de refrigeração diferentes. Também desenvolvemos um novo tipo de frigorífico. Também desenvolvemos um novo tipo de frigorífico. |
| **High-resource models** | |
| Spec-pt-BR | Oferecemos um nmero considerável de modelos de geladeira diferentes. Também desenvolvemos um novo tipo de geladeira. Isso inclui o estilo americano lado a lado refrigeradores. |
| Spec-pt-PT | Oferecemos um número considerável de modelos de frigorífico diferentes. Também desenvolvemos um novo tipo de frigorfico. Estes incluem frigoríficos americanos lado a lado. |
| pt-BR (M-C3_L) | Oferecemos um número considerável de diferentes modelos de geladeira. Também desenvolvemos um novo tipo de geladeira. Estes incluem estilo americano lado a lado, geladeiras. |
| pt-PT (M-C3_L) | Oferecemos um número considerável de diferentes modelos frigoríficos. Também desenvolvemos um novo tipo de frigorfico. Estes incluem estilo americano lado a lado frigoríficos. |

Table 7: English to Portuguese translation generated by Google Translate (as of 20th July, 2018) and translations into Brazilian and European Portuguese generated by our semi-supervised multilingual (M-C2 and M-C3_L) and supervised Spec models. For the underlined English terms both their Brazilian and European translation variants are shown.

trained under unsupervised, supervised, and semi-supervised training data regimes. We reported experimental results on the translation from English to four pairs of language varieties with systems trained under low-resource conditions. We show that in the supervised regime, best performance is achieved by training a multilingual NMT system. For the semi-supervised regime, we compared different automatic labeling strategies that permit to train multilingual neural MT systems with performance comparable to the best supervised NMT system. Our findings were also confirmed by large scale experiments performed on English to Brazilian and European Portuguese. In this scenario, we have also shown that multilingual NMT fully trained on automatic labels can perform very similarly to its supervised version.

In future work, we plan to extend our approach to language varieties in the source side, as well as investigate the possibility of applying transfer-learning (Zoph et al., 2016; Nguyen and Chiang, 2017) for language varieties by expanding our Ada adaptation approach.

## Acknowledgments

This work has been partially supported by the EC-funded project ModernMT (H2020 grant agreement no. 645487). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. Moreover, we thank the Erasmus Mundus European Program in Language and Communication Technology.

## References

Kemal Altintas and lyas Çiçekli. 2002. A Machine Translation System Between a Pair of Closely Related Languages. In *Proceedings of the 17th International Symposium on Computer and Information Sciences (ISCIS 2002)*, pages 192–196.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2018. Neural versus phrase-based mt quality: An in-depth analysis on english-german and english-french. *Computer Speech & Language*, 49:52–70.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit$^3$: Web inventory of transcribed and translated talks. In *Proceedings of the $16^{th}$ Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Marta R Costa-jussà. 2017. Why Catalan-Spanish Neural Machine Translation? Analysis, comparison and combination with standard Rule and Phrase-based technologies. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 55–62.

Marta R Costa-Jussà, Marcos Zampieri, and Santanu Pal. 2018. A Neural Approach to Language Variety Translation. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 275–282.

Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *ACL (1)*, pages 1723–1732.

Nadir Durrani, Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2010. Hindi-to-urdu machine translation through transliteration. In *Proceedings of the 48th Annual meeting of the Association for Computational Linguistics*, pages 465–474. Association for Computational Linguistics.

Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. *arXiv preprint arXiv:1601.01073*.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating Similar Languages: Evaluations and Explorations. In *Proceedings of Language Resources and Evaluation (LREC)*, pages 1800–1807.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Salima Harrat, Karima Meftouh, and Kamel Smaili. 2017. Machine translation for Arabic dialects (survey). *Information Processing & Management*, pages 1–12.

Hany Hassan, Mostafa Elaraby, and Ahmed Y Tawfik. 2017. Synthetic Data for Neural Machine Translation of Spoken-Dialects. In *Proceedings of the 14th International Workshop on Spoken Language Translation*.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2018. Automatic Language Identification in Texts: A Survey.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 427–431.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 4, pages 388–395.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubeši, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating Between Similar Languages and Arabic Dialect Identification: A Report on the Third DSL Shared Task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–14.

Luis Marujo, Nuno Grazina, Tiago Luis, Wang Ling, Luisa Coheur, and Isabel Trancoso. 2011. BP2EP - Adaptation of Brazilian Portuguese texts to European Portuguese. In *Proceedings of the 15th International Conference of the European Association for Machine Translation (EAMT)*, May, pages 129–136.

Toan Q Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 296–301.

Maja Popović, Mihael Arcan, and Filip Klubička. 2016. Language Related Issues for Machine Translation between Closely Related South Slavic Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 43–52.

Nima Pourdamghani and Kevin Knight. 2017. Deciphering Related Languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2503–2508.

Wael Salloum, Heba Elfardy, Linda Alamir-Salloum, Nizar Habash, and Mona Diab. 2014. Sentence Level Dialect Identification for Machine Translation System Selection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 772–778.

Tien-Ping Tan, Sang-Seong Goh, and Yen-Min Khaw. 2012. A Malay Dialect Translation and Synthesis System: Proposal and Preliminary System. In *2012 International Conference on Asian Language Processing*, pages 109–112. IEEE.

Jörg Tiedemann and Nikola Ljubeši. 2012. Efficient Discrimination Between Closely Related Languages. In *Proceedings of COLING 2012: Technical Papers*, pages 2619–2634.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Ronald Wardhaugh. 2006. *An Introduction to Sociolinguistcs*. Blackwell Publishing.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubeši, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial Evaluation Campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–15.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Nikola Ljube. 2014. A Report on the DSL Shared Task 2014. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, 2013, pages 58–67.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the DSL Shared Task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, 2014, pages 1–9.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.

# Effective Parallel Corpus Mining using Bilingual Sentence Embeddings

**Mandy Guo[a]*, Qinlan Shen[b]†*, Yinfei Yang[a]*, Heming Ge[a], Daniel Cer[a],**
**Gustavo Hernandez Abrego[a], Keith Stevens[a], Noah Constant[a],**
**Yun-Hsuan Sung[a], Brian Strope[a], Ray Kurzweil[a]**

[a]Google AI
Mountain View, CA, USA

[b]Carnegie Mellon University
Pittsburgh, PA, USA

## Abstract

This paper presents an effective approach for parallel corpus mining using bilingual sentence embeddings. Our embedding models are trained to produce similar representations exclusively for bilingual sentence pairs that are translations of each other. This is achieved using a novel training method that introduces hard negatives consisting of sentences that are not translations but have some degree of semantic similarity. The quality of the resulting embeddings are evaluated on parallel corpus reconstruction and by assessing machine translation systems trained on gold vs. mined sentence pairs. We find that the sentence embeddings can be used to reconstruct the United Nations Parallel Corpus (Ziemski et al., 2016) at the sentence-level with a precision of 48.9% for en-fr and 54.9% for en-es. When adapted to document-level matching, we achieve a parallel document matching accuracy that is comparable to the significantly more computationally intensive approach of Uszkoreit et al. (2010). Using reconstructed parallel data, we are able to train NMT models that perform nearly as well as models trained on the original data (within 1-2 BLEU).

## 1 Introduction

Volumes of quality parallel training data are critical to neural machine translation (NMT) systems. While large distributed systems have proven useful for mining parallel documents (Uszkoreit et al., 2010; Antonova and Misyurev, 2011), these approaches are computationally intensive and rely on heavily engineered subsystems. Recent work has approached the problem by training lightweight end-to-end models based on word and sentence-level embeddings (Grégoire and Langlais, 2017; Bouamor and Sajjad, 2018; Schwenk, 2018). We propose a novel method for training bilingual sentence embeddings that proves useful for sentence-level mining of parallel data. Sentences are encoded using Deep Averaging Networks (DANs) (Iyyer et al., 2015), a simple bag of n-grams architecture that has been shown to provide surprisingly competitive performance on a number of tasks including sentence classification (Iyyer et al., 2015; Cer et al., 2018), conversation input-response prediction (Yang et al., 2018), and email response prediction (Henderson et al., 2017). Separate encoders are used for each language with candidate source and target sentences being paired based on the dot-product of their embedded representations. Training maximizes the dot-product score of sentence pairs that are translations of each other at the expense of sampled negatives. We contrast using random negatives with carefully selected hard negatives that challenge the model to distinguish between true translation pairs versus non-translation pairs that exhibit some degree of semantic similarity.

The efficiency of the sentence encoders and the use of a dot-product operation to score candidate sentence pairs is well suited for parallel corpus mining. Efficient encoders reduce the amount of computational resources required to obtain sentence embeddings for a large collection of unpaired sentences. Once the sentence embeddings are available, efficient nearest neighbour search (Vanderkam et al., 2013; Johnson et al., 2017) can be used to identify candidate translation pairs.

The language pairs English-French (en-fr) and English-Spanish (en-es) are used in our experiments. Our results show that introducing hard negative sentence pairs, which are semantically similar but that are not translations of each other, systematically outperforms using randomly selected negatives. Our method can be used to reconstruct the United Nations Parallel Corpus (Ziemski et al., 2016) at the sentence-level with a level of precision of 48.9% P@1 for en-fr and 54.9% P@1 for en-es. When we adapt our method to document-

* equal contribution
† Work done during an internship at Google AI.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 165–176
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64017

level pairings we achieve a matching accuracy that is comparable to that of the much heavier weight and more computationally intensive approach of Uszkoreit et al. (2010). Training an NMT model using the reconstructed corpus results in models that perform nearly as well as those trained on the original parallel corpus (within 1-2 BLEU). Finally, our method has a modest degree of correlation with the pair quality scores provided by Zipporah (Xu and Koehn, 2017). However, our method has higher agreement with human judgments, and our approach to filter the ParaCrawl corpus results in NMT systems with higher BLEU scores.

## 2 Approach

This section introduces our bilingual sentence embedding model and the translation candidate ranking task we use for training. We then present our method for selecting hard negative sentence pairs that are not translations of each other but have some degree of semantic similarity. Finally, we detail the use of our bilingual sentence embeddings to search for sentences that are translations of each other, as well as an adaptation to the matching process to parallel documents.

### 2.1 Translation Candidates Ranking Task

Given a pair of sentences that are translations of each other $x$ and $y$, the translation candidate ranking task attempts to rank the true translation $y$ over all other sentences, $\mathcal{Y}$, in the given language. This can be accomplished by modeling the translation probability distribution $P(y \mid x)$. Provided with a scoring function $\phi$ that assesses the compatibility between $x$ and $y$, the distribution can be expressed as the following log-linear model:

$$P(y \mid x) = \frac{e^{\phi(x,y)}}{\sum_{\bar{y} \in \mathcal{Y}} e^{\phi(x,\bar{y})}} \tag{1}$$

To avoid summing over all possible target sentences, the normalization term is approximated by summing over the compatibility score for matching $x$ to $K-1$ sampled negatives together with the compatibility score for the positive candidate:

$$P_{approx}(y \mid x) = \frac{e^{\phi(x,y)}}{\sum_{k=1}^{K} e^{\phi(x,y_k)}} \tag{2}$$

This formulation is similar to early work on discriminative training of log-linear translation decoding models (Och and Ney, 2002). However, rather than using a weighted sum of manually engineered features, we define $\phi$ to be the dot-product of sentence embeddings for the source, $\mathbf{u}$, and target, $\mathbf{v}$, with $\phi(x,y) = \mathbf{u}^\top \cdot \mathbf{v}$. A similar log-linear sentence embedding based formulation of $P(y|x)$ has been previously used for conversation and e-mail response prediction (Henderson et al., 2017; Yang et al., 2018).

### 2.2 Bilingual Sentence Embeddings

Bilingual sentence embeddings are obtained using the dual-encoder architecture illustrated in Figure 1. We use Deep Averaging Networks (DANs) (Iyyer et al., 2015) to compute sentence-level embedding vectors by first averaging word and bi-gram level embeddings, denoted as $\Psi(x)$ and $\Psi(y)$, for the source and target sentences, respectively. [1] The word and bi-gram level embeddings are not pretrained but are rather learned during training of the sentence encoders. The averaged representation is provided to a feedforward deep neural network (DNN). Across hidden layers we include residual connections with a skip level of 1. The final bilingual sentence embeddings are $\mathbf{u}$ and $\mathbf{v}$, which are taken from the last layer of the source and target encoders, respectively. The dot-product of the sentence embeddings, $\mathbf{u}^T \cdot \mathbf{v}$, is used to compute the translation score, $\phi(x, y)$.

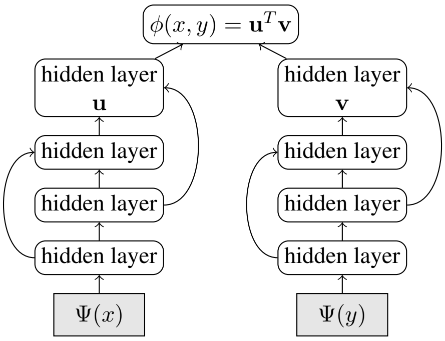


Figure 1: Dual-encoder architecture, where a group of hidden layers encodes source sentence $x$ to $\mathbf{u}$ and a separate group encodes target sentence $y$ to $\mathbf{v}$ such that the score $\phi(x,y)$ is the dot-product $\mathbf{u}^T \cdot \mathbf{v}$.

The dual-encoders are trained for the translation candidate ranking task by maximizing the log likelihood of $P_{approx}$. This objective is particularly

[1] Our implementation sums the word and bi-gram embeddings and then divides the result by $sqrt(n)$, where $n$ is the sentence length. The intuition behind dividing by $sqrt(n)$ is as follows: We want our input embeddings to be sensitive to length. However, we also want to ensure that, for short sequences, the relative differences in the representations are not dominated by sentence length effects.

| | Source (Target) | | Negatives |
|---|---|---|---|
| en-fr | How to display and access shared files<br>(Comment afficher et accéder aux fichiers partagés) | **Random** | Sa respiration devient laborieuse<br>Benoit Faucon Lieu London |
| | | **Hard** | Accès  l’environment des fichiers partagés<br>Des éléments comme des fichiers de dossiers |
| | The General Delegation for Armaments<br>(La délégation générale pour l’armement) | **Random** | RCS 871, où le juge Fauteux explique<br>Avis sur les hôtels |
| | | **Hard** | La 9e armée , commandée par le général Foch<br>La délégation militaire hongroise composée de ... |
| en-es | Oil and gas investments<br>(Inversiones en petróleo y gas) | **Random** | Alquiler mensual desde : 890 USD<br>¿Qué más se deja para preguntar? |
| | | **Hard** | Petróleo y gas<br>Petróleo y Gas Petroquímica página |
| | In Spain, it has clearly chosen the gratuity<br>(En España, se ha elegido claramente la gratuidad) | **Random** | Ve el perfil completo de Fleishman<br>León de montaña en roca |
| | | **Hard** | Dejar propina es una costumbre chilena<br>Este es un típico restaurante español de España |

Table 1: Example of random negatives and hard negatives for en-fr and en-es.

well suited for mini-batch training. As illustrated in Figure 2, within a batch, each source and target translation pair serves as a positive example for that particular pairing with alternative pairings within the same batch treated as negative examples. Given an ordered collection of embeddings for source and target translation pairs, all of the dot-product scores necessary to compute $P_{approx}$ can be determined using a single matrix multiplication of the encoding matrices, $\mathbf{U}$ and $\mathbf{V}^\top$.[2] After the matrix multiplication the scores assigned to true translation pairs can be found on the diagonal while the scores for incorrect pairings are off-diagonal.

Within our experiments, models differ in their selection of the $K-1$ sampled negatives. Our preliminary models make use of the random sampling strategy that has been proven successful in prior work (Henderson et al., 2017; Yang et al., 2018). Using this strategy consists of randomly composing batches of translation pairs and using the matrix multiplication approach described above to obtain within batch negatives for each incorrect pairing We employ random shuffling during training resulting in different random negatives for each $\mathbf{u}_i$ across epochs. As described below we also explore introducing additional hard negatives. This is achieved by extending the target embeddings matrix $\mathbf{V}$ with the sentence embeddings for the hard negatives, which introduces additional off-diagonal values within the matrix of dot-product scores.

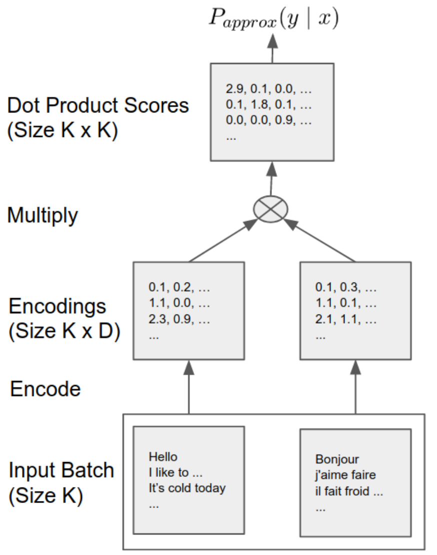


Figure 2: Matrix multiply trick for dot product model with random sampling.

## 2.3 Semantically Similar Hard Negatives

As illustrated in Table 1, randomly selected negatives result in many pairings that are obviously incorrect without requiring a careful assessment of whether the source and randomly sampled targets are true translations. Within a mini-batch, the model could likely achieve a reasonable level of performance by simply identifying which source and target sentences are on the same topic or are otherwise semantically related. However, when mining for parallel data, extracting sentence pairs that are not translations of each other but that are rather merely topically related is expected to harm downstream MT systems that are trained on the erroneous pairs. Given the increased sensitivity of NMT models to data quality issues, perfor-

[2] The encoding matrices are composed of the ordered sentence embeddings for all of the source and target sentences within a batch, $\mathbf{U} = (\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_{k-1})$ and $\mathbf{V} = (\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_{k-1})$.

mance might even be harmed by including semantically similar sentences with sufficient differences in meaning between them. [3]

We improve the mining of true translation pairs by making model training more challenging through the introduction of *hard negatives* – semantically similar translations that are close but not quite identical to the correct translation. The hard negatives are selected using a baseline model trained with randomly sampled negatives. For each source sentence, we identify $M$ hard negatives with target embeddings that achieve high dot-product scores with the source sentence embedding but that are not the correct translation. Examples of hard negatives extracted using the baseline model are provided in Table 1. Compared to the random negatives, hard negatives are semantically more similar to the correct target translation.

As described above, the hard negatives are appended to the target embedding matrix $\mathbf{V}$. Therefore, instead of training with $K$ candidates, each translation input will be compared with $K+K*M$ candidates, where $K$ is the batch size. In practice, getting hard negatives for the entire dataset is very time consuming. We only obtain hard negatives for 20% of the data and use random negative sampling for the remainder of the training set.

## 2.4 Mining Parallel Data

One approach to mining parallel data with bilingual sentence-level embeddings is to independently pair individual source and target candidates based on the similarity of their embeddings. Prior work that explored this approach found that the resulting mined sentence pairs produced poor BLEU scores when used for MT training unless they were combined with traditional human translated corpora with known alignments (Schwenk, 2018). We explore both sentence-level and document-level mining of parallel corpora. For document-level mining, we introduce a novel selection criterion that takes into account the confidence of sentence alignments within a document and sentence position information.

### 2.4.1 Document Matching

Parallel documents are identified as follows: For a given source document, we first run an approximate nearest neighbor (ANN) search for each sentence in the document. This gives us $N$ target sentences for each source sentence (ranked in order of closest match). Let $Y$ be the bag of all target sentences that appear as a match for at least one source sentence. Then for each sentence in $Y$, we look up the document from which they came. We score each candidate document using Eq 3.[4] This scoring function takes into account the sentence-level nearest neighbor rank of the match for source sentence $x$ to target sentence $y$ in the document being scored, $r(x, y)$. The match rank is linearly combined with a normalized confidence score, $f_1(x, y)$, for the match between $x$ and $y$ as well as the absolute difference between the sentence position index of the source and target sentences, $f_2(x, y)$. The sum of the scoring terms is weighted by the hyperparameters, $w_1$ and $w_2$.

$$\sum_{y \in D \cap Y} -r(x,y) + w_1 * f_1(x,y) + w_2 * f_2(x,y) \tag{3}$$

### 2.4.2 Calibrated Confidence Score

The raw dot product score, $\phi(x, y)$, is a poor choice for the confidence score, $f_1(x, y)$. The score from $\phi(x, y)$ provides a relative metric of a translated sentence's match quality with respect to the source sentence, but it is not a globally consistent measurement of how good a translation pair is. Scores are not necessarily in the same range nor do they have comparable relative values for different input source sentences. As a result, if we choose $\phi(x, y)$ to score confidence, there is no single threshold we can use to filter out bad results.

In order to obtain more consistent confidence scores, we propose a novel score normalization model based on dynamic scaling and shifting of the dot product scores. As illustrated in Figure 3, the dynamic scaling and shifting values are computed from the source embedding, $\mathbf{u}$, and a point-wise squaring of the values within the source embedding, $\mathbf{u}^2$. The vectors $\mathbf{u}$ and $\mathbf{u}^2$ are concatenated. The scale and bias terms are computed as

[3] e.g., adding or removing important details according to the sentence similarity scale proposed by Agirre et al. (2012).

[4] Selecting the target document that appears the most in $Y$ should give us a rough idea of which target document is most likely to be the translation of a source document. However, this approach is quite naive since we are ignoring many pieces of information: 1. The rank at which each target sentence appeared, 2. The dot product score between the target sentence and the source sentence, and 3. The indices of the target sentence and the source sentence (i.e. the position of the sentences within their respective documents). Since the first two factors indicate the model's confidence in the sentence match, it is desirable to incorporate this information into our scoring of document matches.

a weighted sum of the concatenated vectors values. After the dynamic scaling and bias terms are used to calibrate the dot-product score, the resulting calibrated dot-product is passed to a sigmoid in order to obtain a final confidence value between 0 and 1. The weights used to compute the scale and bias terms are trained on held out supervised data.

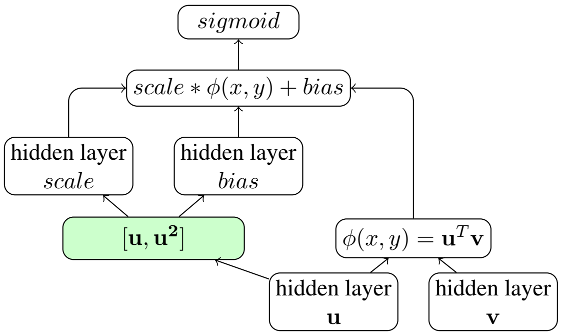

$sigmoid$
$scale * \phi(x, y) + bias$
hidden layer $scale$
hidden layer $bias$
$[\mathbf{u}, \mathbf{u^2}]$
$\phi(x, y) = \mathbf{u}^T\mathbf{v}$
hidden layer $\mathbf{u}$
hidden layer $\mathbf{v}$

Figure 3: Scoring model based on dual-encoder architecture.

It is worth noting that because the hidden layers for $scale$ and $bias$ only use features from the source embeddings, it will not affect the ranking of targets. Thus, we still always use dot-product similarity, $\phi(x, y)$, to retrieve targets via nearest neighbor search. For document-level matching, we convert the dot-product values into the calibrated confidence scores, $f_1(x, y)$, without needing to reinspect the target embeddings.

## 3 Experiments

We train our proposed model on two language pairs: English-French (en-fr) and English-Spanish (en-es). First, we evaluate the performance on the translation candidate ranking task, comparing the dual-encoder architectures with random negative sampling versus using hard negatives. Then, we present results for document-level matching using Uszkoreit et al. (2010)'s method as a strong baseline. We explore training NMT systems using our method to both filter and re-construct parallel corpora. Finally, we assess the level or agreement between our method and human judgments.

### 3.1 Data

For training the model, we construct a parallel corpus using a system similar to the approach described in Uszkoreit et al. (2010). The final constructed corpus contains around 600M en-fr sentence pairs and 470M en-es sentence pairs. To assess the quality of the parallel corpus, we ask human annotators to manually evaluate the constructed pairs. The human annotators judge whether 200 randomly selected sentence pairs for both en-fr and en-es are GOOD or BAD translations. We find that the GOOD translation rate is around 80% for both language pairs. The constructed parallel corpus is split into two parts: a training set (90%) and a held-out dev set (10%), with the held-out dev set being used for our preliminary reconstruction experiments.

The UN corpus (Ziemski et al., 2016) is used for additional corpus reconstruction experiments. The corpus consists of 800k manually translated UN documents from 1990 to 2014 for the six official UN languages. 86k of these documents are fully aligned at the sentence-level for all 15 language pairs. We make use of the fully aligned en-fr and en-es document pairs and extract all aligned sentence pairs from those document pairs. There are a total of 11.3 million aligned sentence pairs each for en-fr and en-es. Assuming that we have no knowledge about which documents and sentences are aligned, the task is to reconstruct the document and sentence pairs.

We evaluate trained translation models on wmt13 (Bojar et al., 2013) and wmt14 (Bojar et al., 2014) for en-es and en-fr, respectively. Translation models are trained using data taken from the parallel corpus described above that was constructed using Uszkoreit et al. (2010)'s method. Additional translation experiments make use of ParaCrawl[5], a dataset containing 4 billion noisy translation pairs for en-fr and 2 billion pairs for en-es. Within Paracrawl, each pair contains pre-computed scores by Zipporah (Xu and Koehn, 2017) and the Bicleaner tool, which estimates the translation quality of the pair. We make use of the Zipporah scores to compare translation models trained on filtered versions of the corpus selected using Zipporah versus our method.

### 3.2 Experimental Configuration

Model configuration and hyperparameters for our sentence embedding models are set mostly based on defaults taken from prior work with very minimal tuning on the held-out dev set. For each language, we build a vocabulary consisting of 200 thousands unigram and 200 thousands bi-gram tokens. All inputs are tokenized and normalized be-

[5] https://paracrawl.eu

fore being fed to the model. We employ an SGD optimizer with a batch size of 128. The learning rate is set to 0.01 with a learning decay of 0.96 every 5 million steps. We train for 50 million steps.

For each encoder layer, we employ a four-layer DNN model which contains 320, 320, 500 and 500 hidden units for each layer respectively. We apply a ReLU activation in the first three layers and no activation in the final layer. We enable residual connections between layers with a skip level of 1. There is no parameter sharing between the source and target encoder layers. The size of the unigram and bi-gram embeddings is set to 320 and the embeddings are updated during the training process. The sentence embedding size is set to 512 for both source and target languages.

The calibrated confidence score is trained jointly with the translation candidate ranking task but with a stop gradient that prevents the confidence task from modifying the bilingual sentence encoders. The tasks are trained in a multitask framework with multiple workers, where 90% of the workers optimize the translation candidate ranking task and the remaining 10% optimize the confidence task. We use the same configuration for confidence as when training the translation candidate ranking task. Both use the same batch size 128, meaning there is 1 positive and 127 negative candidates selected for each pass over an example. We apply a dropout of 0.4 before feeding the feature vector $[\mathbf{u}, \mathbf{u^2}]$ into the hidden layers that calculate $scale$ and $bias$.

### 3.3 Dev Set Sentence-level Matching

We first evaluate the trained models on the translation target retrieval task and use precision at N (P@N) as our evaluation metric. For every source sentence in the dev set, we run the model and find the nearest neighbors from a set of possible target sentences. Previous work (Henderson et al., 2017; Yang et al., 2018) usually evaluated P@N from 100 examples (1 positive and 99 negatives). We find that this does not work well for the translation target ranking task. Rather, the P@N of 100 metric goes up to 99.9% quickly and provides no differentiation between models trained with different configurations.

In this work, we evaluate the P@N from the true target sentence (positive) and 10 million random selected target (negatives) given a source sentence. We score all selected targets using the translation pair scoring model and rank them accordingly. The P@N score evaluates if the true translation target (positive) is in the top N target candidates. We evaluate the model with random sampling and $M$ hard negatives for $M$=5, 10, 20. Recall that the number of negatives is equal to the batch size for the models trained with random sampling. The number of negatives for hard negative models, however, is $K + K * M$ where $K$ is the batch size. To make a fair comparison, we also evaluate a model trained with additional random samples, by augmenting the number of random negatives to $K + K * 20$.

Table 2 shows the P@N results of the proposed models for N=1, 3, 10. The model with random negatives provides a strong baseline for finding the right translation target, with a P@1 metric of 70.49% for en-fr and 67.81% for en-es. The augmented random negative model performs better than the base random negative model for en-es. However, the hard negative models outperform the random negative models across all metrics. Even with only 5 hard negatives, the P@1 metrics improved by 8% for en-fr and 3% for en-es. The addition of more hard negatives, however, does not always further improve performance.

## 4 Reconstructing the United Nations Corpus

In this section, we demonstrate that the proposed model can be used to efficiently reconstruct the United Nations (UN) Parallel Corpus (Ziemski et al., 2016).

### 4.1 UN Sentence-level Matching

We first apply the dual-encoder model to mine target candidates at the sentence-level. As mentioned in section 1, one of the advantages of the dual-encoder model is that it is straightforward to use it to encode the source and target sentences separately. Taking advantage of this property, we first pre-encode all target sentences into a target database, and then we iterate through the source sentences to retrieve the potential targets for each one of them using an approximated nearest neighbour (ANN) search (Vanderkam et al., 2013). The target sentence retrieval pipeline using ANN search is shown in Figure 4.

Once again we first use P@N as the evaluation metric for target retrieval, for N=1, 3, 10. We evaluate the two random sampling models and a

| Negative Selection Approach | en-fr |  |  | en-es |  |  |
|---|---|---|---|---|---|---|
|  | P@1 | P@3 | P@10 | P@1 | P@3 | P@10 |
| Random Negatives | 70.49 | 80.03 | 86.39 | 67.81 | 77.37 | 84.42 |
| Random Negatives (Augmented) | 70.67 | 79.99 | 86.14 | 70.47 | 79.79 | 86.33 |
| (5) Hard Negatives | 78.31 | 85.30 | 89.52 | 73.46 | 82.37 | 87.75 |
| (10) Hard Negatives | 77.06 | 84.04 | 88.70 | 74.92 | 83.29 | 88.14 |
| (20) Hard Negatives | 78.29 | 85.06 | 89.58 | 74.84 | 82.86 | 88.23 |

Table 2: Precision at N (P@N) results on the evaluation set for models built using the random negatives and $(M)$ hard negatives. Models attempt to select the true translation target for a source sentence against 10M randomly selected targets.

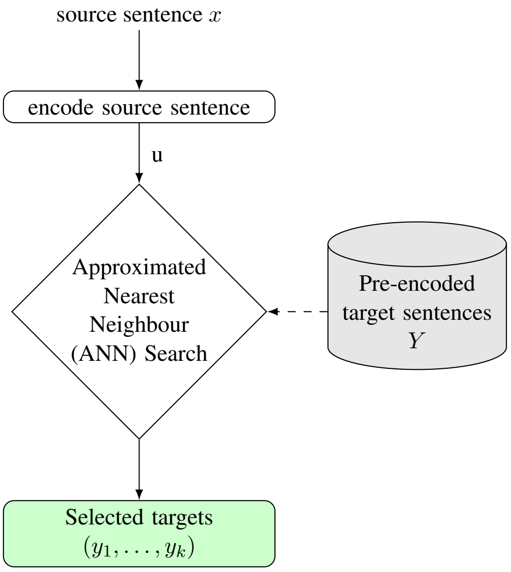

source sentence $x$
encode source sentence
u
Approximated Nearest Neighbour (ANN) Search
Pre-encoded target sentences $Y$
Selected targets $(y_1, \dots, y_k)$

Figure 4: Target sentence retrieval pipeline.

hard negative model with 20 hard negatives for each example. As shown in table 3, with random negatives, the P@1 metric is 34.83% for en-fr and 44.89% for en-es. Adding hard negatives boosts the performance on all metrics, improving the P@1 metric more than 10% absolute in both en-fr and en-es – 48.9% for en-fr and 54.9% for en-es.

### 4.2 UN Document-level Matching

In our final reconstruction experiment, we make use of the document-level matching method outlined in section 2.4.1. The hyperparameters $N$, $w_1$, and $w_2$ are set to 10, 5, and $-2$, respectively, based on prior experiments with the translation matching task on the dev set. We compare using the document matching score proposed by Eq. (3) to scoring document pairs by counting the number of Viterbi aligned sentences linking the two together. As a strong baseline, we also include the application of Uszkoreit et al. (2010)'s method to the UN dataset.

Table 4 shows the document matching accuracies. Using Eq. (3) to score document matches outperforms counting mutually aligned sentences. Moreover, while our approach is simpler and less computationally intensive than Uszkoreit et al. (2010)'s, it obtains a promising level of performance.

## 5 Evaluation Using a Translation Model

As a proof of concept on using our mined translation pairs as training data, we train translation models with original versus mined parallel sentence pairs from UN corpus, and with filtered ParaCrawl data using Zipporah score versus using our model's confidence score. We evaluate on wmt13 (Bojar et al., 2013) and wmt14 (Bojar et al., 2014) testing sets for en-es and en-fr, respectively, with performance assessed using BLEU (Papineni et al., 2002).

The translation models are based on Transformer architecture (Vaswani et al., 2017), and make use of a model dimension of 512 and a hidden dimension of 2048, with 6 layers and 8 attention heads. The models use the Adam optimizer with the training schedule described in Vaswani et al. (2017). For each language pair, sentence pairs are segmented using a shared 32,000 wordpiece vocabulary (Schuster and Nakajima, 2012). Sentence pairs are then batched together by approximate sequence length with variable batch sizes based on sequence length. The average batch size per step is 120 pairs per batch. We train each model until convergence (approximately 120K steps).

| Negative Selection Approach | en-fr | | | en-es | | |
|---|---|---|---|---|---|---|
| | P@1 | P@3 | P@10 | P@1 | P@3 | P@10 |
| Random Negative | 34.83 | 47.99 | 61.20 | 44.89 | 58.13 | 70.36 |
| Random Negative (Augmented) | 36.51 | 49.07 | 61.37 | 47.08 | 59.55 | 71.34 |
| (20) Hard Negative | 48.90 | 62.26 | 73.03 | 54.94 | 67.78 | 78.06 |

Table 3: Precision at N (P@N) of target sentence retrieval on the UN corpus. Models attempt to select the true translation target for a source sentence from the entire corpus (11.3 million aligned sentence pairs.)

| Matching method | en-fr | en-es |
|---|---|---|
| Alignment Counts | 82.1 | 85.1 |
| Our approach Eq. (3) | 89.0 | 90.4 |
| Uszkoreit et al. (2010) | 93.4 | 94.4 |

Table 4: Accuracy of document matching on UN corpus.

### 5.1 Mined UN Corpus

We compare translation models trained on the reconstructed UN corpora for en-fr and en-es with models trained on the original UN pairs, which we use as Oracle models.

We examine two versions of the reconstructed corpora. In the first version, we take the highest scoring match at the sentence-level as the mined parallel sentence pairs and these pairs are then filtered by their calibrated confidence score[6] with default threshold 0.5. In the second version, we perform document-level matching over the UN dataset. Within paired documents, we follow Uszkoreit et al. (2010) and employ a dynamic programming sentence alignment algorithm informed by sentence length and multilingual probabilistic dictionaries. In both versions, we drop sentence pairs where both sides are either identical or a language detector declares them to be in the wrong language. As a post-processing step, the resulting translations are resegmented using the Moses tokenizer and true-cased before evaluation (Koehn et al., 2007).

Table 5 shows the results obtained from the models trained on the different variations of the parallel data. The models trained with mined pairs perform very close to the Oracle model, demonstrating the effectiveness of the proposed parallel corpus mining approach. Training on the mined sentence-level pairs even does slightly better than using the Oracle data for en-es. This is presumably because the mined pairs are cleaner due to the filtering step. We notice, however, that training on the UN corpus gives translation results that are much lower than the state-of-the-art on the WMT evaluation sets. This is likely due to the fact that the UN parallel corpus is small and drawn from a particularly restricted domain.

| | **en-fr (wmt14)** | **en-es (wmt13)** |
|---|---|---|
| Mined sentence-level | 29.63 | 29.03 |
| Mined document-level | 30.05 | 27.09 |
| Oracle | 30.96 | 28.81 |

Table 5: BLEU scores on WMT testing sets of the NMT models trained on original UN pairs (Oracle) and on two versions of mined UN corpora.

| | **en-fr (wmt14)** | **en-es (wmt13)** |
|---|---|---|
| WMT | 38.38 | 32.69 |
| Our data | 39.81 | 33.75 |
| Zipporah | 39.29 | 33.58 |
| WMT + Our data | 40.30 | 34.15 |
| WMT + Zipporah | 39.29 | 34.07 |

Table 6: BLEU scores on WMT testing sets of the NMT models trained on different data: 1) WMT training sets, 2) filtered ParaCrawl data, and 3) combined data of WMT and filtered ParaCrawl.

### 5.2 Filtered ParaCrawl data

We compare the performance of training translation models[7] on ParaCrawl data filtered using Zipporah scores versus our scoring method. For this experiment, our confidence score is fine-tuned on the ParaCrawl corpus using an additional 900k positive and 900k negative examples selected based on having extreme Zipporah

[6] The confidence model is trained with a dev set which consist of 1/10 of UN corpora, these data are removed from training.

[7] Using the same model parameters as earlier experiments.

scores.[8] With Zipporah, we select all examples from ParaCrawl with a Zipporah score greater than or equal to 0, which is the threshold used in the official release. There are 43 million such pairs in en-fr and 24 million in en-es. We then select the same number of pairs from the ParaCrawl data that have the highest scores from our fine-tuned model. As illustrated in Table 6, the performance achieved by the ParaCrawl trained models on the WMT test data is quite high, both achieves better performance comparing with the baseline model trained on WMT training set. This suggests that filtered ParaCrawl data is a good source of general-purpose training material. Models trained on our filtered data slightly outperform those trained on data filtered by Zipporah. Row 4 and 5 also show the performance of models trained on the combined data of WMT and our filtered ParaCrawl and combined data of WMT and Zipporah filtered data respectively[9]. Combining the datasets further improves the translation performance about 0.5 blue score, and model trained on WMT and our filtered ParaCrawl data achieves the best performance.

### 5.3 Qualitative Analysis of Filtered ParaCrawl Data

On the ParaCrawl corpus we find that the Pearson's $r$ between Zipporah and our calibrated confidence scores is only $0.4$. This correlation is quiet low given the level of translation performance achieved by both methods when they are used to select training pairs for an NMT system and suggests that the two methods may provide complementary information.

We access the agreement of the two methods on extreme score values.[10] We sample a balanced data set consisting of 100k pairs with extreme positive Zipphora values and 100k pairs with extreme negative values. At a threshold of 0.5 and without an fine-tuning, our method agrees with the extreme Zipporah scores with an accuracy of 78.2% for en-fr and 80.5% for en-es. However, using the confidence scores fine-tuned to ParaCrawl from section 5.2, we achieve a high level of agreement of 98.4% for en-fr and 98.6% with fine-tuning.

|  | en-fr | en-es |
|---|---|---|
| zipporah | 72.0 | 74.0 |
| our model | 76.0 | 74.5 |

Table 7: GOOD translation rate (%) annotated by translation professionals.

We perform an evaluation using human judgments comparing our scoring model against Zipporah scores on the ParaCrawl data. As in the filtering experiments, we select all examples from ParaCrawl with a Zipporah score greater than or equal to zero and then select a matching number of pairs with the highest scores from our model. We then sample 200 examples from each set and send them to translation professionals for evaluation. Each example is examined by one annotator that labels the pair as either a GOOD or BAD translation. A GOOD translation means more than 70% of a sentence is correctly translated in the paired sentences, meaning most of the information is conveyed.

Table 7 shows the GOOD translation rate for each sampled subset. The performance between the two approaches is close for en-es and the proposed score normalization model is 4% better for en-fr. In our analysis of the BAD translation pairs, one common failure pattern from the proposed model is that one of the sentences is only partially translated in the other sentence. This is likely because we are still missing enough of these types of hard negatives in the training data. We also find our model produces more pairs where the sentences on both sides are identical. These identical pairs are mostly labeled as BAD translations because they are unlikely to be actual translations.

## 6 Related Work

The problem of obtaining high-quality parallel corpora, or bitexts, is one of the most critical issues in machine translation. One longstanding approach for extracting parallel corpora is to mine documents from the web (Resnik, 1999). Much of the previous work on parallel document mining has relied on using metadata, such as document titles (Yang and Li, 2002), publication dates (Munteanu and Marcu, 2005, 2006) or document structure (Chen and Nie, 2000; Resnik and Smith, 2003; Shi et al., 2006), to identify bitexts.

[8] Extreme positive score values from Zipporah are considered to be those in the top 1% of the agreement scores found in the ParaCrawl corpus. Extreme negative score values are considered to be agreement scores in the bottom 50% of the Zipporah scores for ParaCrawl.

[9] The sizes of WMT training set and filtered ParaCrawl are very close, so we simply mix the data together without any up sampling or down sampling.

[10] For this analysis we use the same definition of extreme Zipporah scores as in section 5.2

Another direction, however, is to identify bitexts using only textual information, as the metadata associated with documents can often be sparse or unreliable (Uszkoreit et al., 2010). Some text-based approaches for identifying bitexts rely on methods such as n-gram scoring (Uszkoreit et al., 2010), named entity matching (Do et al., 2009), and cross-language information retrieval (Utiyama and Isahara, 2003; Munteanu and Marcu, 2005).

There is active research on using embedding-based approaches where texts are mapped to an embedding space in order to determine whether they are bitexts. Grégoire and Langlais (2017) use a Siamese network (Yin et al., 2015) to map source and target language sentences into the same space, then classify whether the sentences are parallel based on labelled data. Hassan et al. (2018) obtain English and Chinese sentence embeddings in a shared space by averaging encoder states from a bilingual shared encoder NMT system. The cosine similarity between these sentence embeddings is then used as a measure of cross-lingual similarity between the sentences, which can then be used to filter out noisy sentence pairs. Schwenk (2018) use a similar approach but learn a joint embedding over nine languages. Our model differs from previous approaches, as it uses a dual-encoder architecture instead of an encoder-decoder architecture. Not only is the dual-encoder architecture is more efficient (Henderson et al., 2017), it also allows us to directly train toward extracting parallel sentences from a collection of candidates.

## 7 Conclusion

In this paper, we present an effective parallel corpus mining approach using sentence embeddings produced by a bilingual dual-encoder model. The proposed model encodes source sentences and target sentences into sentence embeddings separately and then calculates the dot-product score for these two embedding vectors to assess translation pair quality. We propose the selection of hard negatives that consist of semantically similar sentence pairs that are not translations of each other. Our experiments reveal that using hard negatives improves the ability of our model to identify true translation pairs. We find the proposed method to be useful for both mining and filtering parallel data. Our method compares favorably to Zipporah for filtering, while for mining it provides a lightweight alternative to Uszkoreit et al. (2010)'s method.

## Acknowledgement

We thank our teammates from Descartes, Translate and other Google groups for their feedback and suggestions. Special thanks goes to Wei Wang and Melvin Johnson for extracting translation data and discussions on NMT models.

## References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Alexandra Antonova and Alexey Misyurev. 2011. Building a web-based parallel corpus and filtering out machine-translated text. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 136–144. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58.

Houda Bouamor and Hassan Sajjad. 2018. H2@bucc18: Parallel sentence extraction from comparable corpora using multilingual sentence embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Jiang Chen and Jian-Yun Nie. 2000. Parallel web text mining for cross-language ir. In *Content-Based Multimedia Information Access-Volume 1*, pages 62–77. Centre de Hautes Etudes Internationale D'Informatique Documentaire.

Thi-Ngoc-Diep Do, Viet-Bac Le, Brigitte Bigi, Laurent Besacier, and Eric Castelli. 2009. Mining a comparable text corpus for a vietnamese-french statistical machine translation system. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 165–172. Association for Computational Linguistics.

Francis Grégoire and Philippe Langlais. 2017. A deep neural network approach to parallel sentence extraction. *arXiv preprint arXiv:1709.09783*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting parallel sub-sentential fragments from nonparallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 81–88. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 295–302, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Philip Resnik. 1999. Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 527–534. Association for Computational Linguistics.

Philip Resnik and Noah A Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

M. Schuster and K. Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Holger Schwenk. 2018. Filtering and mining parallel data in a joint multilingual space. *arXiv preprint arXiv:1805.09822*.

Lei Shi, Cheng Niu, Ming Zhou, and Jianfeng Gao. 2006. A dom tree alignment model for mining parallel data from the web. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 489–496. Association for Computational Linguistics.

Jakob Uszkoreit, Jay M. Ponte, Ashok C. Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1101–1109, Stroudsburg, PA, USA. Association for Computational Linguistics.

Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 72–79. Association for Computational Linguistics.

Dan Vanderkam, Rob Schonberger, Henry Rowley, and Sanjiv Kumar. 2013. Nearest neighbor search in google correlate. Technical report, Google.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950.

Christopher C Yang and Kar Wing Li. 2002. Mining english/chinese parallel documents from the world wide web. In *Proceedings of the 11th International World Wide Web Conference, Honolulu, USA*.

Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. In *The 3rd Workshop on Representation Learning for NLP (RepL4NLP)*, Melbourne, Australia. Association for Computational Linguistics.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, LREC '16. European Language Resources Association.

# On The Alignment Problem In Multi-Head Attention-Based Neural Machine Translation

**Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney**
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
`<surname>@i6.informatik.rwth-aachen.de`

## Abstract

This work investigates the alignment problem in state-of-the-art multi-head attention models based on the transformer architecture. We demonstrate that alignment extraction in transformer models can be improved by augmenting an additional alignment head to the multi-head source-to-target attention component. This is used to compute sharper attention weights. We describe how to use the alignment head to achieve competitive performance. To study the effect of adding the alignment head, we simulate a dictionary-guided translation task, where the user wants to guide translation using pre-defined dictionary entries. Using the proposed approach, we achieve up to 3.8% BLEU improvement when using the dictionary, in comparison to 2.4% BLEU in the baseline case. We also propose alignment pruning to speed up decoding in alignment-based neural machine translation (ANMT), which speeds up translation by a factor of 1.8 without loss in translation performance. We carry out experiments on the shared WMT 2016 English→Romanian news task and the BOLT Chinese→English discussion forum task.

## 1 Introduction

Attention-based neural machine translation (NMT) (Bahdanau et al., 2015) uses an attention layer to determine which part of the input sequence to focus on during decoding. This component eliminates the need for explicit alignment modeling. In conventional phrase-based statistical machine translation (Koehn et al., 2003), word alignment is modeled explicitly, making it clear which word or phrase is being translated. The lack of explicit alignment use in attention-based models makes it harder to determine which target words are generated using which source words. While this is not necessarily needed for translation itself, alignments can be useful in certain applications, e.g. when the customer wants to enforce specific translation of certain words.

One simple solution is to use maximum attention weights to extract the alignment, but this can result in wrong alignments in the case where the maximum attention weight is not pointing to the word being translated. Such cases are not uncommon in NMT, making the use of attention weights as alignment replacement non-trivial (Chatterjee et al., 2017; Hasler et al., 2018). Alignment extraction is even less clear for transformer models (Vaswani et al., 2017), which currently produce state-of-the-art results. These models use multiple attention components for each of the stacked decoder layers. In this work we focus our study on these models since they usually outperform single-attention-head recurrent neural network (RNN) attention models.[1]

Alignment-based NMT (Alkhouli et al., 2016) uses neural models trained using explicit hard alignments to generate translation. These systems include explicit alignment modeling, making them more convenient for tasks where the source-to-target alignment is needed. However, it is not clear whether these systems are able to compete with strong attention-based NMT systems. Alkhouli and Ney (2017) present results for alignment-based neural machine translation (ANMT) using models trained on CPUs, limiting them to small models of 200-node layers, and they only investigate RNN models. Wang et al. (2018) present results using only one RNN encoder layer, and do not include attention layers in their models. In this work, we investigate the performance of large and deep state-of-the-art transformer models. We keep the multi-head attention component and propose to augment it with an additional alignment head, to

[1] The transformer models won in most of the WMT 2018 news translation tasks: `http://matrix.statmt.org`.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 177–185
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64018

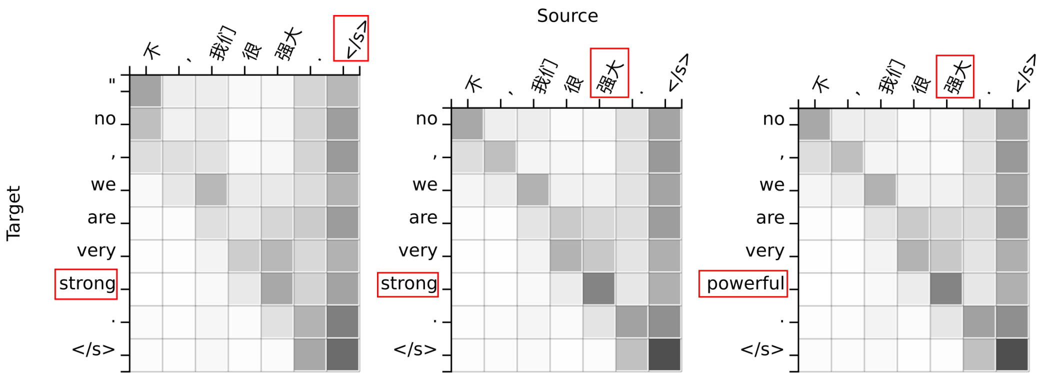


Figure 1: An example from the Chinese→English system. The figures illustrate the accumulated attention weights of the baseline transformer model (left), the alignment-assisted transformer model (middle), and the alignment-assisted model guided by a dictionary entry. We simulate a scenario where the user wants to translate the Chinese word "强大" to "powerful". Both the baseline and alignment-assisted transformer models generate the translation "strong" instead. To enforce the translation, we use the maximum attention weight to determine the source word being translated. Left: The maximum attention of the baseline case incorrectly points to the sentence end when translating the designated Chinese word, therefore we cannot enforce the translation in this case. Middle: The alignment looks sharper because the system has an augmented alignment head. In this case the maximum attention is pointing to the correct Chinese word. Right: using the maximum attention, the translation "strong" is successfully replaced with the translation "powerful" as suggested by the user using our proposed alignment-assisted transformer.

combine the benefits of the two. We demonstrate that we can train these models to achieve competitive results in comparison to strong state-of-the-art baselines. Moreover, we demonstrate that this variant has clear advantage in tasks that require alignments such as dictionary-guided translation.

Translation in NMT can be performed without explicit alignment. However, there are tasks where translation needs to be constrained given specific user requirements. Examples include interactive machine translation, and scenarios where customers demand domain-specific words or phrases to be translated according to a pre-defined dictionary. We demonstrate that the explicit use of alignment in ANMT can be leveraged to generate guided translation. Figure (1) illustrates an example. The figures are generated using attention weights averaged over all attention components in each system.

The contribution of this work is as follows. First, we propose a method to integrate alignment information into the multi-head attention component of the transformer model (Section 3.1). We describe how such models can be trained to maintain the strong baseline performance while also using external alignment information (Section 3.3). We also introduce alignment models that use self-attentive layers for faster evaluation (Section 3.2). Second, we introduce alignment pruning during search to speed up evaluation without affecting translation quality (Section 4). Third, we describe how to extract alignments from multi-head attention models (Section 5), and demonstrate that alignment-assisted transformer systems perform better than baseline systems in dictionary-guided translation tasks (Section 7). We present speed and performance results in Section 6.

## 2 Related Work

Alignment-based neural models have explicit dependence on the alignment information either at the input or at the output of the network. They have been extensively and successfully applied on top of conventional phrase-based systems (Sundermeyer et al., 2014; Tamura et al., 2014; Devlin et al., 2014). In this work, we focus on using the models directly to perform standalone neural machine translation.

Alignment-based neural models were proposed in (Alkhouli et al., 2016) to perform neural machine translation. They mainly used feedforward alignment and lexical models in decoding. Alkhouli and Ney (2017) used recurrent models instead, and presented an attention component biased using external alignment information. In this work, we explore the use of transformer models in ANMT instead of recurrent models.

Deriving neural models for translation based on the hidden Markov model (HMM) framework can also be found in (Yang et al., 2013; Yu et al., 2017). Alignment-based neural models were also applied to perform summarization and morphological inflection (Yu et al., 2016). Their work used a monotonous alignment model, where training was done by marginalizing over the alignment hidden variables, which is computationally expensive. In this work, we use non-monotonous alignment models. In addition, we train using pre-computed Viterbi alignments which speeds up neural training. In (Yu et al., 2017), alignment-based neural models were used to model alignment and translation from the target to the source side (inverse direction), and a language model was included in addition. They showed results on a small translation task. In this work, we present results on translation tasks containing tens of millions of words. We do not include a language model in any of our systems.

There is plenty of work on modifying attention models to capture more complex dependencies. Cohn et al. (2016) introduce structural biases from word-based alignment concepts like fertility and Markov conditioning. These are internal modifications that leave the model self-contained. Our modifications introduce alignments as external information to the model. Arthur et al. (2016) include lexical probabilities to bias attention. Chen et al. (2016) and Mi et al. (2016) add an extra term dependent on the alignments to the training objective function to guide neural training. This is only applied during training but not during decoding. Our work makes use of alignments during training and also during decoding.

There are several approaches to perform constrained translation. One possibility is including this information in training, but this requires knowing the constraints at training time (Crego et al., 2016). Post-processing the hypotheses is another possibility, but this comes with the downside that offline modification of the hypotheses happens out of context. A third possibility is to do constrained decoding (Hokamp and Liu, 2017; Chatterjee et al., 2017; Hasler et al., 2018; Post and Vilar, 2018). This does not require knowledge of the constraints at training time, and it also allows dynamic changes of the rest of the hypothesis when the constraints are activated. We perform experiments where the translation is guided online during decoding. We focus on the case where translation suggestions are to be used when a word in the source sentence matches the source side of a pre-defined dictionary entry. We show that alignment-assisted transformer-based NMT outperforms standard transformer models in such a task.

## 3 Alignment-Based Neural Machine Translation

Alignment-based NMT divides translation into two steps: (1) alignment and (2) word generation. The system is composed of an alignment model and a lexical model that can be trained jointly or separately. During translation, the alignment is hypothesized first, and the lexical score is computed next using the hypothesized alignment (Alkhouli et al., 2016). Hence, each translation hypothesis has an underlying alignment used to generate it. The alignment model scores the alignment path.

Formally, given a source sentence $f_1^J = f_1...f_j...f_J$, a target sentence $e_1^I = e_1...e_i...e_I$, and an alignment sequence $b_1^I = b_1...b_i...b_I$, where $j = b_i \in \{1, 2, ..., J\}$ is the source position aligned to the target position $i \in \{1, 2, ..., I\}$, we model translation using an alignment model and a lexical model:

$$\begin{aligned} p(e_1^I|f_1^J) &= \sum_{b_1^I} p(e_1^I, b_1^I|f_1^J) \qquad (1) \\ &\approx \max_{b_1^I} \prod_{i=1}^{I} \underbrace{p(e_i|b_i, b_1^{i-1}, e_1^{i-1}, f_1^J)}_{\text{lexical model}} \cdot \\ &\qquad \underbrace{p(b_i|b_1^{i-1}, e_1^{i-1}, f_1^J)}_{\text{alignment model}} . \end{aligned}$$

Both the lexical model and the alignment model have rich dependencies including the full source context $f_1^J$, the full alignment history $b_1^{i-1}$, and the full target history $e_1^{i-1}$. The lexical model has an extra dependence on the current source position $b_i$.

While previous work focused on RNN structures for the lexical and alignment models (Alkhouli and Ney, 2017), we use multi-head self-attentive transformer model structures instead. The next two subsections describe the structural details of these models.

### 3.1 Transformer-Based Lexical Model

In this work we propose to use lexical models based on the transformer architecture (Vaswani et al., 2017). This architecture has the following main components:

- self-attentive layers replacing recurrent layers. These layers are parallelizable due to the lack of sequential dependencies that recurrent layers have.

- multi-head source-to-target attention: several attention heads are used to attend to the source side. Each attention head computes a normalized probability distribution over the source positions. The attention heads are concatenated. Each decoder layer in the model has its own multi-head attention component.

We propose to condition the lexical model on the alignment information. We add a special alignment head

$$\alpha(j|b_i) = \begin{cases} 1, & \text{if} \quad j = b_i \\ 0, & \text{otherwise.} \end{cases}$$

defined for the source positions $j, b_i \in \{1, 2, ..., J\}$. This is a one-hot distribution that has a value of 1 at position $j$ that matches the aligned position $b_i$. This head is then concatenated to the rest of the attention heads as shown in Figure (2). The one-hot alignment distribution is used similar to attention weights to weight the encoded source representations, effectively selecting the representation $h_{b_i}$ which corresponds to the aligned word.

### 3.2 Self-Attentive Alignment Model

In this work we use self-attentive layers instead of RNN layers in the alignment model. This removes the sequential dependency of computing RNN activations and allows for parallelization. We replace the bidirectional RNN encoder of the alignment model by multi-head self-attentive layers as described in (Vaswani et al., 2017). We also use multi-head self-attentive layers to replace the RNN layers in the decoder part of the network. There are two main differences when comparing this self-attentive alignment model to the transformer architecture described in (Vaswani et al., 2017). (1) The output is a probability distribution over possible source jumps $\Delta_i = b_i - b_{i-1}$, that is, the model predicts the likelihood of jumping from the previous source position $b_{i-1}$ to the current source position $b_i$. (2) There is no multi-head source-to-target attention layer as in the transformer network. Rather, we use a single-head hard attention layer. This layer is not computed like attention weights, but it is constructed using the previous alignment point $b_{i-1}$ using

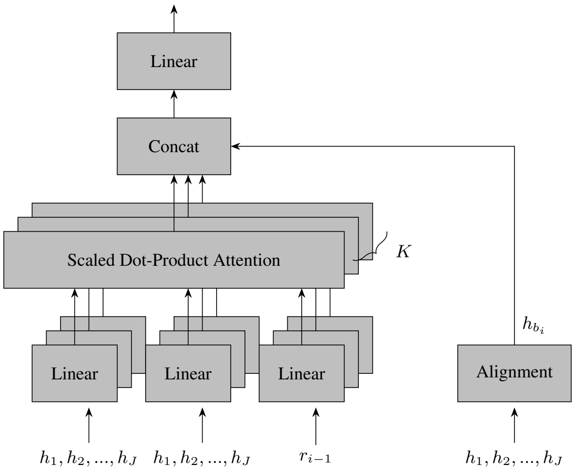


Figure 2: Alignment-assisted multi-head attention component. $h_1, h_2, ..., h_J$: the encoder states at all $J$ source positions, $h_{b_i}$: the encoder state at the aligned source position $b_i$, $r_{i-1}$: the previous decoder state, $K$: number of attention heads. Removing the alignment block results in the default multi-head source-to-target attention component of (Vaswani et al., 2017).

$$\alpha(j|b_{i-1}) = \begin{cases} 1, & \text{if} \quad j = b_{i-1} \\ 0, & \text{otherwise.} \end{cases}$$

defined for the source positions $j, b_{i-1} \in \{1, 2, ..., J\}$. When multiplied by the source encodings, $\alpha$ effectively selects the source encoding $h_{b_{i-1}}$ of the previous aligned position. This is then summed up with the decoder state $r_{i-1}$.

### 3.3 Training

Our attempts to train the alignment-assisted transformer lexical model from scratch achieved suboptimal results. This could happen because the model could choose to over-rely on the alignment information, risking that the remaining attention heads would become useless, especially during the early stages of training. To overcome this, we first trained the transformer baseline parameters without the alignment information until convergence, and used the trained parameters to initial-

**Algorithm 1** Alignment-Based Pruned Decoding

```
1:  procedure TRANSLATE($f_1^J$, $beamSize$, $threshold$)
2:      $hyps \leftarrow initHyp$          ▷init. set of partial hypotheses
3:      while GETBEST($hyps$) not terminated do
4:          ▷compute alignment distribution in batch mode
5:          $alignDists \leftarrow$ALIGNMENTDIST($hyps$)
6:          ▷hypothesize source alignment points
7:          $activePos \leftarrow \{\}$
8:          for $pos$ From $1$ to $J$ do
9:              ▷position computed if at least one
10:             ▷beam entry surpasses the threshold
11:             for $b$ From $1$ to $beamSize$ do
12:                 if $alignDists[b, pos] > threshold$ then
13:                     $activePos.Append(pos)$
14:                     break
15:         ▷evaluate all positions if none survived pruning
16:         if $activePos$ is empty then
17:             $activePos \leftarrow \{1, ...J\}$
18:         ▷compute lexical distributions of all
19:         ▷hypotheses in hyps in batch mode
20:         $lexDists \leftarrow$ LEXICALDIST($hyps$, $activePos$)
21:         ▷combine lexical and alignment scores
22:         $hyps \leftarrow Combine(lexDists, alignDists)$
23:         ▷prune to fit the beam
24:         $hyps \leftarrow Prune(hyps, beamSize)$
25:     ▷return the best scoring hypothesis
26:     return GETBEST($hyps$)
```

ize the alignment-assisted model training. This resulted in better systems compared to training from scratch. We were able to see significant perplexity improvements in the second stage of training indicating that the model was making use of the newly introduced information. Further details are discussed in Section 6.1.

## 4 Alignment Pruning

Alignment-based decoding requires hypothesizing alignment positions in addition to word translations. The algorithm is shown in Algorithm (1). Each lexical hypothesis has an underlying alignment hypothesis ($activePos$) that is used to compute it (line 20). This is done as a part of beam search. To speed up decoding, we compute the alignment model output first for all beam entries (line 5). This gives a distribution over the next possible source positions. We prune all source positions that have a probability below a fixed $threshold$ (lines 12–14 ). We only evaluate the lexical model for those positions that survive the threshold. If the pruning threshold is too aggressive to let any of the source positions survive, pruning is disabled for that time step (lines 16–17).

## 5 Alignment Extraction

We use attention weights to extract the alignments at each time step during decoding. We look up the source word having the maximum accumulated attention weight

$$j(i) = \underset{\hat{j} \in \{1 \ldots J\}}{\operatorname{argmax}} \left\{ \sum_{l=1}^{L} \sum_{k=1}^{K} \alpha_{i,k,l}(\hat{j}) \right\}$$

where $K$ is the number of attention heads per decoder layer, $L$ is the number of decoder layers, $\alpha_{i,k,l}(\hat{j})$ is the attention weight at source position $\hat{j} \in \{1, ..., J\}$ for target position $i$ of the $k$-th head computed for the the $l$-th decoder layer. This is an extension of using maximum attention weights in single-head attention models (Chatterjee et al., 2017). In the alignment-assisted transformer, the aligned position is given by:

$$j(i, j') = \underset{\hat{j} \in \{1 \ldots J\}}{\operatorname{argmax}} \left\{ \sum_{l=1}^{L} \left( \sum_{k=1}^{K} \alpha_{i,k,l}(\hat{j}) + \alpha(\hat{j}|j') \right) \right\}$$

where $j' \in \{1, ..., J\}$ is the hypothesized source position during search, and $\alpha(\hat{j}|j')$ is the alignment indicator which is equal to 1 if $\hat{j} = j'$ and zero otherwise. This effectively gives a preference for the hypothesized position over all other positions. Note that the hypothesized positions are scored during translation using the alignment model described in Section 3.2.

## 6 Experiments

We run experiments on the WMT 2016 English→Romanian news task,[2] and on BOLT Chinese→English which is a discussion forum task. The corpora statistics are shown in Table (1).

All transformer models use 6 encoder and 6 decoder self-attentive layers. We use 8 scaled dot product attention heads and augment an additional alignment head to the source-to-target attention component. We use an embedding size of 512. The size of feedforward layers is 2048 nodes. We use source and target weight tying for the WMT English→Romanian task, and no tying for BOLT Chinese→English.

The structure of the RNN models is as follows. The English→Romanian lexical and alignment models use 1 bidirectional encoder layer. The

[2] http://www.statmt.org/wmt16/

|  | WMT 2016 |  | BOLT |  |
|---|---|---|---|---|
|  | **English** | **Romanian** | **Chinese** | **English** |
| `Train` sentence pairs | 604K |  | 4.1M |  |
| `Train` running words | 15.5M | 15.8M | 80M | 88M |
| `Dev` sentence pairs | 1000 |  | 1845 |  |
| `Test` sentence pairs | 1999 |  | 1124 |  |
| Vocabulary | 92K | 128K | 380K | 815K |
| Neural network vocabulary | 50K | 50K | 50K | 50K |

Table 1: Corpora statistics.

|  |  |  | WMT En→Ro<br>newstest2016 |  |  | BOLT Zh→En<br>test |  |  |
|---|---|---|---|---|---|---|---|---|
| # | System | Layer size | PPL | BLEU [%] | TER [%] | PPL | BLEU [%] | TER [%] |
| baselines |  |  |  |  |  |  |  |  |
| 1 | Attention baseline | 1000 | 10.2 | 24.7 | 58.9 | 8.0 | 20.0 | 65.6 |
| 2 | Transformer baseline | 2048 | 6.2 | 27.9 | 54.6 | 6.0 | 22.5 | 62.1 |
| 3 | (Alkhouli and Ney, 2017) | 200 | - | 24.8 | 58.1 | - | - | - |
| this work |  |  |  |  |  |  |  |  |
| 4 | RNN Attention align.-biased | 1000 | 7.2 | 26.4 | 56.1 | 5.6 | 19.6 | 62.3 |
| 5 | Align.-assisted Transformer | 2048 | **5.0** | **28.1** | **54.3** | **4.7** | **22.7** | **61.8** |

Table 2: Translation results for the WMT 2016 English→Romanian task and the BOLT Chinese→English task. We include the lexical model perplexities.

Chinese→English models have 1 bidirectional encoder and 3 stacked unidirectional encoder layers. All models use 2 decoder layers. The baseline attention models have similar structures. We use LSTM layers of 1000 nodes and embeddings of size 620. We train using the Adam optimizer (Kingma and Ba, 2015). All alignment models predict source jumps of maximum width of 100 source positions (forward and backward).

The alignments used during training are the result of IBM1/HMM/IBM4 training using GIZA++ (Och and Ney, 2003). All results are measured in case-insensitive BLEU[%] (Papineni et al., 2002). TER[%] scores are computed with *TERCom* (Snover et al., 2006). We implement the models in Sockeye (Hieber et al., 2017), which allows efficient training of large models on GPUs.

### 6.1 Performance Comparison

Table (2) presents results on the two tasks. The RNN attention (row 1) and transformer (row 2) baselines are shown. The transformer baseline outperforms the attention baseline by a large margin. We also include the English→Romanian system of Alkhouli and Ney (2017). This is an alignment-based RNN attention system which uses 200-node layers. We also trained our own alignment-based RNN attention system using larger layers of 1000 nodes. This is shown in row 4. Our RNN system outperforms the previously published alignment-based results (row 3) by 1.6% BLEU and 2.0% TER. This is due to the increase in model size.

Our proposed alignment-assisted transformer system is shown in row 5. This system outperforms the RNN alignment-based system of row 4 by 1.7% BLEU on the English→Romanian task, establishing a new state-of-the-art result for alignment-based neural machine translation. We also achieve 3.1% BLEU improvement over our RNN alignment-biased attention system on the Chinese→English task. In comparison to the transformer baseline (row 2), the proposed system achieves similar performance on both tasks. We compare the development perplexity to check whether the lexical model makes use of the alignment information. Indeed, the baseline transformer development perplexity drops from 6.2 to 5.0 on English→Romanian and from 6.0 to 4.7

|  |  | WMT En→Ro |  |  | BOLT Zh→En |  |  |
|---|---|---|---|---|---|---|---|
| # | Alignment | #entries | BLEU$^{[\%]}$ | TER$^{[\%]}$ | #entries | BLEU$^{[\%]}$ | TER$^{[\%]}$ |
| 1 | Transformer baseline | - | 27.3 | 55.6 | - | 24.2 | 61.5 |
| 2 | + dictionary | 3.1K | 29.7 | 55.4 | 4.6K | 25.5 | 61.0 |
| 3 | Alignment-assisted Transformer | - | 27.2 | 55.5 | - | 24.2 | 60.8 |
| 4 | + dictionary | 3.1K | **31.0** | **53.0** | 4.6K | **26.4** | **58.6** |

Table 3: Improvements after using the dictionary of the development sets. The tokenized references of the English→Romanian and Chinese→English development sets have 26.7K and 46.6K running words respectively.

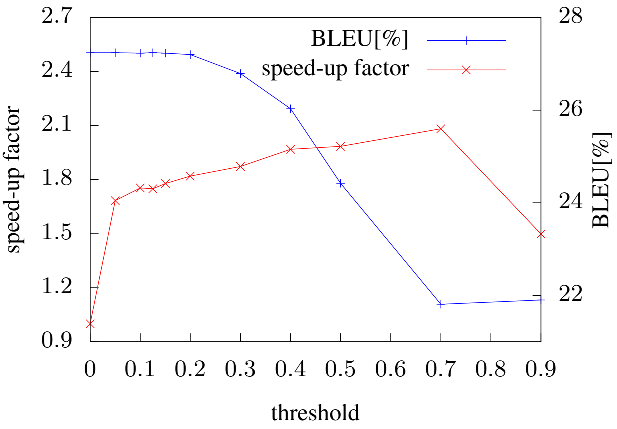


Figure 3: Speed up and translation quality in BLEU vs. pruning threshold on the WMT English→Romanian task.

on Chinese→English, indicating that the model is making use of the alignment information.

### 6.2 Decoding Speed Up

Figure (3) shows the speed-up factor and performance in BLEU over different threshold values. The speed-up factor is computed against the no-pruning case (i.e. threshold 0). The batch size used in these experiments is 5. We speed up translation by a factor of 1.8 without loss in translation quality at threshold 0.15. Higher threshold values result in more aggressive pruning and hence a degradation in translation quality. It is interesting to note that at threshold 0.05 we achieve a speed up of 1.7, implying that significant pruning happens at low threshold values. At high threshold values, speed starts to go down, since we have more cases where no alignment points survive the threshold, in which case pruning is disabled as discussed in Algorithm (1, lines 16–17).

## 7 Dictionary Suggestions

We evaluate the use of attention weights as alignments in a dictionary suggestion task, where a predefined dictionary of suggested one-to-one translations is given. We perform a relaxed form of constrained translation, i.e. we do not ensure that the suggestion will make it to the translation. To this end, we use attention weights to extract the alignments at each time step during decoding as described in Section 5. We look up the source word $f_{j(i)}$ having the maximum accumulated attention weight in the dictionary. If the word matches the source-side of a dictionary entry, we enforce the translation to match the dictionary suggestion $e(f_{j(i)})$ by setting an infinite cost for all but the suggested word.

We create a simulated dictionary using the reference side of the development set. We map the reference to the source words using IBM4 alignment. The development set is concatenated with the training data to obtain good-quality alignment. We exclude English stop words,[3] and only use source words aligned one-to-one to target words. We include up to 4 dictionary entries per sentence, and add reference translations only if they are not part of the baseline (i.e. unconstrained) translation, similar to (Hasler et al., 2018).

Table (3) shows results for the dictionary suggestions task described in Section (7). The English→Romanian dictionary covers 11.6% of the reference set, while the Chinese→English dictionary has 9.9% coverage. We observe larger improvement when using the dictionary entries in the alignment-assisted transformer system in comparison to the transformer baseline systems. Our system improves BLEU by 3.8%, while the baseline is improved only by 2.4% BLEU on the English→Romanian task. We also observe larger

[3] Long stop list: `https://www.ranks.nl/stopwords`

improvements in the Chinese→English case. This suggests that the maximum attention weights in alignment-assisted systems can point more accurately to the word being translated, allowing the use of more dictionary entries. As shown in Figure (1), the accumulated attention weights are sharper when the system has an augmented alignment head. This explains the larger improvements our systems achieve.

## 8 Conclusion

We proposed augmenting transformer models with an alignment head to help extract alignments in scenarios such as dictionary-guided translation. We demonstrated that the alignment-assisted systems can achieve competitive performance compared to strong transformer baselines. We also showed that the alignment-assisted systems outperformed standard transformer models when used for dictionary-guided translation on two tasks. Finally, we achieved a speed-up factor of 1.8 by pruning alignment hypotheses in alignment-based decoding while maintaining translation quality. In future work we plan to investigate alternative pruning methods like histogram pruning. We also plan to investigate the performance of alignment-assisted transformer models in constrained decoding settings, where the user demands specific translation of certain words.

## Acknowledgments

European Research Council
Established by the European Commission

This research has received funding from the European Research Council (ERC) (under the European Union's Horizon 2020 research and innovation programme, grant agreement No 694537, project "SEQCLAS") and the Deutsche Forschungsgemeinschaft (DFG; grant agreement NE 572/8-1, project "CoreTec"). Tamer Alkhouli was partly funded by the 2016 Google PhD fellowship for North America, Europe and the Middle East. The work reflects only the authors' views and none of the funding parties is responsible for any use that may be made of the information it contains.

## References

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. 2016. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 54–65, Berlin, Germany.

Tamer Alkhouli and Hermann Ney. 2017. Biasing attention-based recurrent neural networks using external alignment information. In *EMNLP 2017 Second Conference on Machine Translation*, pages 108–117, Copenhagen, Denmark.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Austin, Texas.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, San Diego, Calefornia, USA.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frédéric Blain. 2017. Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*, pages 157–168, Copenhagen, Denmark.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. In *Proceedings of the 2016 Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 121–134, Austin, Texas.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, et al. 2016. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *52nd Annual Meeting of the Association for Computational Linguistics*, pages 1370–1380, Baltimore, MD, USA.

Eva Hasler, Adrià De Gisper, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 506–512, New Orleans, Louisiana, USA.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, San Diego, Calefornia, USA.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-03)*, pages 127–133, Edmonton, Alberta.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2283–2288, Austin, Texas.

Franz J. Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. 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*, pages 223–231, Cambridge, Massachusetts, USA.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. 2014. Translation Modeling with Bidirectional Recurrent Neural Networks. In *Conference on Empirical Methods on Natural Language Processing*, pages 14–25, Doha, Qatar.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2014. Recurrent neural networks for word alignment model. In *52nd Annual Meeting of the Association for Computational Linguistics*, pages 1470–1480, Baltimore, MD, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Weiyue Wang, Derui Zhu, Tamer Alkhouli, Zixuan Gan, and Hermann Ney. 2018. Neural hidden markov model for machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 377–382. Association for Computational Linguistics.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word alignment modeling with context dependent deep neural network. In *51st Annual Meeting of the Association for Computational Linguistics*, pages 166–175, Sofia, Bulgaria.

Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomás Kociský. 2017. The neural noisy channel. In *Proceedings of the International Conference on Learning Representations*, volume abs/1611.02554.

Lei Yu, Jan Buys, and Phil Blunsom. 2016. Online segment to segment neural transduction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1307–1316, Austin, Texas.

# A Call for Clarity in Reporting BLEU Scores

**Matt Post**
Amazon Research
Berlin, Germany

## Abstract

The field of machine translation faces an under-recognized problem because of inconsistency in the reporting of scores from its dominant metric. Although people refer to "the" BLEU score, BLEU is in fact a parameterized metric whose values can vary wildly with changes to these parameters. These parameters are often not reported or are hard to find, and consequently, BLEU scores between papers cannot be directly compared. I quantify this variation, finding differences as high as 1.8 between commonly used configurations. The main culprit is different tokenization and normalization schemes applied to the reference. Pointing to the success of the parsing community, I suggest machine translation researchers settle upon the BLEU scheme used by the annual Conference on Machine Translation (WMT), which does not allow for user-supplied reference processing, and provide a new tool, SACREBLEU,[1] to facilitate this.

## 1 Introduction

Science is the process of formulating hypotheses, making predictions, and measuring their outcomes. In machine translation research, the predictions are made by models whose development is the focus of the research, and the measurement, more often than not, is done via BLEU (Papineni et al., 2002). BLEU's relative language independence, its ease of computation, and its reasonable correlation with human judgments have led to its adoption as the dominant metric for machine translation research. On the whole, it has been a boon to the community, providing a fast and cheap way for researchers to gauge the performance of their models. Together with larger-scale controlled manual evaluations, BLEU has shepherded the field through a decade and a half of quality improvements (Graham et al., 2014).

This is of course not to claim there are no problems with BLEU. Its weaknesses abound, and much has been written about them (cf. Callison-Burch et al. (2006); Reiter (2018)). This paper is not, however, concerned with the shortcomings of BLEU as a proxy for human evaluation of quality; instead, our goal is to bring attention to the relatively narrower problem of the *reporting* of BLEU scores. This problem can be summarized as follows:

- BLEU is not a single metric, but requires a number of parameters (§2.1).
- Preprocessing schemes have a large effect on scores (§2.2). Importantly, BLEU scores computed against differently-processed references are not comparable.
- Papers vary in the hidden parameters and schemes they use, yet often do not report them (§2.3). Even when they do, it can be hard to discover the details.

Together, these issues make it difficult to evaluate and compare BLEU scores across papers, which impedes comparison and replication. I quantify these issues and show that they are serious, with variances bigger than many reported gains. After introducing the notion of *user-* versus *metric-supplied* tokenization, I identify user-supplied reference tokenization as the main cause of this incompatibility. In response, I suggest the community use only *metric-supplied* reference tokenization when sharing scores,[2] following the annual Conference on Machine Translation (Bojar et al., 2017, WMT). In support of this, I release a

[1] `https://github.com/awslabs/sockeye/tree/master/contrib/sacrebleu`

[2] Sometimes referred to as *detokenized BLEU*, since it requires that system output be detokenized prior to scoring.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 186–191
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64019

Python package, SacreBLEU,[3] which automatically downloads and stores references for common test sets, thus introducing a "protective layer" between them and the user. It also provides a number of other features, such as reporting a version string which records the parameters used and which can be included in published papers.

## 2 Problem Description

### 2.1 Problem: BLEU is underspecified

"BLEU" does not signify a single concrete method, but a constellation of parameterized methods. Among these parameters are:

- The number of references used;
- for multi-reference settings, the computation of the length penalty;
- the maximum n-gram length; and
- smoothing applied to 0-count n-grams.

Many of these are not common problems in practice. Most often, there is only one reference, and the length penalty calculation is therefore moot. The maximum n-gram length is virtually always set to four, and since BLEU is corpus level, it is rare that there are any zero counts.

But it is also true that people use BLEU scores as very rough guides to MT performance across test sets and languages (comparing, for example, translation performance into English from German and Chinese). Apart from the wide intra-language scores between test sets, the number of references included with a test set has a large effect that is often not given enough attention. For example, WMT 2017 includes two references for English–Finnish. Scoring the online-B system with one reference produces a BLEU score of 22.04, and with two, 25.25. As another example, the NIST OpenMT Arabic–English and Chinese–English test sets[4] provided four references and consequently yielded BLEU scores in the high 40s (and now, low 50s). Since these numbers are all gathered together under the label "BLEU", over time, they leave an impression in people's minds of very high BLEU scores for some language pairs or test sets relative to others, but without this critical distinguishing detail.

### 2.2 Problem: Different reference preprocessings cannot be compared

The first problem dealt with parameters used in BLEU scores, and was more theoretical. A second problem, that of preprocessing, exists in practice.

Preprocessing includes input text modifications such as normalization (e.g., collapsing punctuation, removing special characters), tokenization (e.g., splitting off punctuation), compound-splitting, the removal of case, and so on. Its general goal is to deliver meaningful white-space delimited tokens to the MT system. Of these, tokenization is one of the most important and central. This is because BLEU is a precision metric, and changing the reference processing changes the set of n-grams against which system n-gram precision is computed. Rehbein and Genabith (2007) showed that the analogous use in the parsing community of $F_1$ scores as rough estimates of cross-lingual parsing difficulty were unreliable, for this exact reason. BLEU scores are often reported as being *tokenized* or *detokenized*. But for computing BLEU, both the system output and reference are always tokenized; what this distinction refers to is whether the reference preprocessing is *user-supplied* or *metric-internal* (i.e., handled by the code implementing the metric), respectively. And since BLEU scores can only be compared when the reference processing is the same, user-supplied preprocessing is error-prone and inadequate for comparing across papers.

Table 1 demonstrates the effect of computing BLEU scores with different reference tokenizations. This table presents BLEU scores where a single WMT 2017 system (online-B) and the reference translation were both processed in the following ways:

- *basic*. User-supplied preprocessing with the Moses tokenizer (Koehn et al., 2007).[5]
- *split*. Splitting compounds, as in Luong et al. (2015a):[6] e.g., *rich-text* → *rich - text*.
- *unk*. All word types not appearing at least twice in the target side of the WMT training data (with "basic" tokenization) are mapped to UNK. This hypothetical scenario could

[3] `pip3 install sacrebleu`

[4] `https://catalog.ldc.upenn.edu/LDC2010T21`

[5] Arguments `-q -no-escape -protected basic-protected-patterns -l LANG`.

[6] Their use of compound splitting is not mentioned in the paper, but only here: `http://nlp.stanford.edu/projects/nmt`.

|  | English→ ⋆ |  |  |  |  |  | ⋆ →English |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| config | en-cs | en-de | en-fi | en-lv | en-ru | en-tr | cs-en | de-en | fi-en | lv-en | ru-en | tr-en |
| basic | 20.7 | 25.8 | 22.2 | 16.9 | 33.3 | 18.5 | 26.8 | 31.2 | 26.6 | 21.1 | 36.4 | 24.4 |
| split | 20.7 | 26.1 | 22.6 | 17.0 | 33.3 | 18.7 | 26.9 | 31.7 | 26.9 | 21.3 | 36.7 | 24.7 |
| unk | 20.9 | 26.5 | 25.4 | 18.7 | 33.8 | 20.6 | 26.9 | 31.4 | 27.6 | 22.7 | 37.5 | 25.2 |
| metric | 20.1 | 26.6 | 22.0 | 17.9 | 32.0 | 19.9 | 27.4 | 33.0 | 27.6 | 22.0 | 36.9 | 25.6 |
| *range* | 0.6 | 0.8 | 0.6 | 1.0 | 1.3 | 1.4 | 0.6 | 1.8 | 1.0 | 0.9 | 0.5 | 1.2 |
| basic$_{lc}$ | 21.2 | 26.3 | 22.5 | 17.4 | 33.3 | 18.9 | 27.7 | 32.5 | 27.5 | 22.0 | 37.3 | 25.2 |
| split$_{lc}$ | 21.3 | 26.6 | 22.9 | 17.5 | 33.4 | 19.1 | 27.8 | 32.9 | 27.8 | 22.2 | 37.5 | 25.4 |
| unk$_{lc}$ | 21.4 | 27.0 | 25.6 | 19.1 | 33.8 | 21.0 | 27.8 | 32.6 | 28.3 | 23.6 | 38.3 | 25.9 |
| metric$_{lc}$ | 20.6 | 27.2 | 22.4 | 18.5 | 32.8 | 20.4 | 28.4 | 34.2 | 28.5 | 23.0 | 37.8 | 26.4 |
| *range$_{lc}$* | 0.6 | 0.9 | 0.5 | 1.1 | 0.6 | 1.5 | 0.7 | 1.7 | 1.0 | 1.0 | 0.5 | 1.2 |

Table 1: BLEU score variation across WMT'17 language arcs for cased (top) and uncased (bottom) BLEU. Each column varies the processing of the "online-B" system output and its references. *basic* denotes basic user-supplied tokenization, *split* adds compound splitting, *unk* replaces words not appearing at least twice in the training data with UNK, and *metric* denotes the metric-supplied tokenization used by WMT. The *range* row lists the difference between the smallest and largest scores, excluding *unk*.

easily happen if this common user-supplied preprocessing were inadvertently applied to the reference.

- *metric*. Only the metric-internal tokenization of the official WMT scoring script, `mteval-v13a.pl`, is applied.[7]

The changes in each column show the effect these different schemes have, as high as 1.8 for one arc, and averaging around 1.0. The biggest is the treatment of case, which is well known, yet many papers are not clear about whether they report cased or case-insensitive BLEU.

Allowing the user to handle pre-processing of the reference has other traps. For example, many systems (particularly before sub-word splitting (Sennrich et al., 2016) was proposed) limited the vocabulary in their attempt to deal with unknown words. It's possible that these papers applied this same unknown-word masking to the references, too, which would artificially inflate BLEU scores. Such mistakes are easy to introduce in researcher pipelines.[8]

| paper | configuration |
|---|---|
| Chiang (2005) | metric$_{lc}$ |
| Bahdanau et al. (2014) | *(unclear)* |
| Luong et al. (2015b) | user or metric *(unclear)* |
| Jean et al. (2015) | user |
| Wu et al. (2016) | user or user$_{lc}$ *(unclear)* |
| Vaswani et al. (2017) | user or user$_{lc}$ *(unclear)* |
| Gehring et al. (2017) | user, metric |

Table 2: Benchmarks set by well-cited papers use different BLEU configurations (Table 1). Which one was used is often difficult to determine.

### 2.3 Problem: Details are hard to come by

User-supplied reference processing precludes direct comparison of published numbers, but if enough detail is specified in the paper, it is at least possible to reconstruct comparable numbers. Unfortunately, this is not the trend, and even for meticulous researchers, it is often unwieldy to include this level of technical detail. In any case, it creates uncertainty and work for the reader. One has to read the experiments section, scour the footnotes, and look for other clues which are sometimes scattered throughout the paper. Figuring out what another team did is not easy.

The variations in Table 1 are only some of the possible configurations, since there is no limit to the preprocessing that a group could apply. But assuming these represent common, concrete configurations, one might wonder how easy it is to determine which of them was used by a particular paper. Table 2 presents an attempt to recover this information from a handful of influential papers in the literature. Not only are systems not comparable due to different schemes, in many cases, no easy determination can be made.

[7] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v13a.pl`

[8] This paper's observations stem in part from an early version of a research workflow I was using, which applied preprocessing to the reference, affecting scores by half a point.

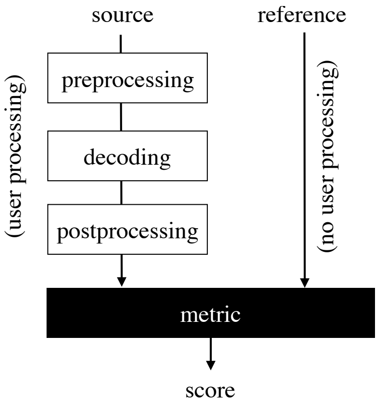


Figure 1: The proper pipeline for computing reported BLEU scores. White boxes denote user-supplied processing, and the black box, metric-supplied. The user should not touch the reference, while the metric applies its own processing to the system output and reference.

### 2.4 Problem: Dataset specification

Other tricky details exist in the management of datasets. It has been common over the past few years to report results on the English→German arc of the WMT'14 dataset. It is unfortunate, therefore, that for this track (and this track alone), there are actually *two* such datasets. One of them, released for the evaluation, has only 2,737 sentences, having removed about 10% of the original data after problems were discovered during the evaluation. The second, released after the evaluation, restores this missing data (after correcting the problem) and has 3,004 sentences. Many researchers are unaware of this fact, and do not specify which version they use when reporting, which itself contributes to variance.

### 2.5 Summary

Figure 1 depicts the ideal process for computing sharable scores. Reference tokenization must identical in order for scores to be comparable. The widespread use of user-supplied reference preprocessing prevents this, needlessly complicating comparisons. The lack of details about preprocessing pipelines exacerbates this problem. This situation should be fixed.

## 3 A way forward

### 3.1 The example of PARSEVAL

An instructive comparison comes from the evaluation of English parsing scores, where numbers have been safely compared across papers for decades using the PARSEVAL metric (Black et al., 1991). PARSEVAL works by taking labeled spans of the form $(N, i, j)$ representing a nonterminal $N$ spanning a constituent from word $i$ to word $j$. These are extracted from the parser output and used to compute precision and recall against the gold-standard set taken from the correct parse tree. Precision and recall are then combined to compute the $\mathrm{F}_1$ metric that is commonly reported and compared across parsing papers.

Computing parser $\mathrm{F}_1$ comes with its own set of hidden parameters and edge cases. Should one count the `TOP` (`ROOT`) node? What about `-NONE-` nodes? Punctuation? Should any labels be considered equivalent? These boundary cases are resolved by that community's adoption of a standard codebase, `evalb`,[9] which included a parameters file that answers each of these questions.[10] This has facilitated almost thirty years of comparisons on treebanks such as the Wall Street Journal portion of the Penn Treebank (Marcus et al., 1993).

### 3.2 Existing scripts

MOSES[11] has a number of scoring scripts. Unfortunately, each of them has problems. Moses' `multi-bleu.perl` cannot be used because it requires user-supplied preprocessing. The same is true of another evaluation framework, MultEval (Clark et al., 2011), which explicitly advocates for user-supplied tokenization.[12] A good candidate is Moses' `mteval-v13a.pl`, which makes use of metric-internal preprocessing and is used in the annual WMT evaluations. However, this script inconveniently requires the data to be wrapped into XML. Nematus (Sennrich et al., 2017) contains a version (`multi-bleu-detok.perl`) that removes the XML requirement. This is a good idea, but it still requires the user to manually handle the reference translations. A better approach is to keep the reference away from the user entirely.

### 3.3 SACREBLEU

SACREBLEU is a Python script that aims to treat BLEU with a bit more reverence:

- It expects detokenized outputs, applying its own metric-internal preprocessing, and produces the same values as WMT;

[9] `http://nlp.cs.nyu.edu/evalb/`

[10] The configuration file, `COLLINS.PRM`, answers these questions as no, no, no, and ADVP=PRT.

[11] `http://statmt.org/moses`

[12] `https://github.com/jhclark/multeval`

- it automatically downloads and stores WMT (2008–2018) and IWSLT 2017 (Cettolo et al., 2017) test sets, obviating the need for the user to handle the references at all; and

- it produces a short version string that documents the settings used.

SacreBLEU can be installed via the Python package management system:

```
pip3 install sacrebleu
```

It can then be used to download the source side of test sets as decoder input—all WMT test sets are available, as well as recent IWSLT test sets, and others are being added. After decoding and detokenization, it can then used to produce BLEU scores.[13] The following command selects the WMT'14 EN-DE dataset used in the official evaluation:

```
cat output.detok \
  | sacrebleu -t wmt14 -l en-de
```

(The restored version that was released after the evaluation (§2.4) can be selected by using `-t wmt14/full`.) It prints out a version string recording all the parameters as '+' delimited KEY.VALUE pairs (here shortened with `--short`):

```
BLEU+c.mixed+l.en-de+#.1+s.exp
  +t.wmt14+tok.13a+v.1.2.10
```

recording:

- mixed case evaluation

- on EN-DE

- with one reference

- and exponential smoothing

- on the WMT14 dataset

- using the WMT standard '13a' tokenization

- with SacreBLEU 1.2.10.

SacreBLEU is open source software released under the Apache 2.0 license.

## 4 Summary

Research in machine translation benefits from the regular introduction of test sets for many different language arcs, from academic, government, and industry sources. It is a shame, therefore, that we are in a situation where it is difficult to directly compare scores across these test sets. One might be tempted to shrug this off as an unimportant detail, but as was shown here, these differences are in fact quite important, resulting in large variances in the score that are often much higher than the gains reported by a new method.

Fixing the problem is relatively simple. Research groups should only report BLEU computed using a metric-internal tokenization and preprocessing scheme for the reference, and they should be explicit about the BLEU parameterization they use. With this, scores can be directly compared. For backwards compatibility with WMT results, I recommend the processing scheme used by WMT, and provide a new tool that makes it easy to do so.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *ArXiv e-prints*, abs/1409.0473.

E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991.*

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214. Association for Computational Linguistics.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluation the role of bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics.*

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stker, Katsuitho Sudoh, Koichiro Yoshino, and Christian Federmann. 2017.

[13] The CHRF metric is also available via the `-m` flag.

Overview of the iwslt 2017 evaluation campaign. In *14th International Workshop on Spoken Language Translation*, pages 2–14, Tokyo, Japan.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. 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*, pages 176–181. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. 2017. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 123–135. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2014. Is machine translation getting better over time? In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 443–451. Association for Computational Linguistics.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10. Association for Computational Linguistics.

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. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015b. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics, Volume 19, Number 2, June 1993, Special Issue on Using Large Corpora: II.*

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Ines Rehbein and Josef van Genabith. 2007. Treebank annotation schemes and parser evaluation for german. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Ehud Reiter. 2018. A structured review of the validity of bleu. *Computational Linguistics*, 0(0):393–401.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv e-prints*, abs/1706.03762.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv e-prints*, abs/1609.08144.

# Exploring Gap Filling as a Cheaper Alternative to Reading Comprehension Questionnaires when Evaluating Machine Translation for Gisting

**Mikel L. Forcada**
Dept. Lleng. i Sist. Inf.,
Universitat d'Alacant
E-03690 St. Vicent del Raspeig,
Spain
mlf@ua.es

**Carolina Scarton,
Lucia Specia**
Dept. of Comput. Sci.,
Univ. of Sheffield,
Sheffield S1 4DP, U.K
{l.specia,c.scarton}
@shef.ac.uk

**Barry Haddow,
Alexandra Birch**
School of Computing,
Univ. Edinburgh,
Edinburgh EH8 4AB, U.K
{bhaddow,a.birch}
@ed.ac.uk

## Abstract

A popular application of machine translation (MT) is *gisting*: MT is consumed *as is* to make sense of text in a foreign language. Evaluation of the usefulness of MT for gisting is surprisingly uncommon. The classical method uses *reading comprehension questionnaires* (RCQ), in which informants are asked to answer professionally-written questions in their language about a foreign text that has been machine-translated into their language. Recently, *gap-filling* (GF), a form of *cloze* testing, has been proposed as a cheaper alternative to RCQ. In GF, certain words are removed from reference translations and readers are asked to fill the gaps left using the machine-translated text as a hint. This paper reports, for the first time, a comparative evaluation, using both RCQ and GF, of translations from multiple MT systems for the same foreign texts, and a systematic study on the effect of variables such as gap density, gap-selection strategies, and document context in GF. The main findings of the study are: (a) both RCQ and GF clearly identify MT to be useful; (b) global RCQ and GF rankings for the MT systems are mostly in agreement; (c) GF scores vary very widely across informants, making comparisons among MT systems hard, and (d) unlike RCQ, which is framed around documents, GF evaluation can be framed at the sentence level. These findings support the use of GF as a cheaper alternative to RCQ.

## 1 Introduction

### 1.1 Machine translation for gisting

Machine translation (MT) applications fall in two main groups: *assimilation* or *gisting*, and *dissemination*. Assimilation refers to the use of the raw MT output to make sense of foreign texts. Dissemination refers to the use of the MT output as a draft translation that can be *post-edited* into a publishable translation. The needs of both groups of applications are quite different; for instance, an otherwise *perfect* Russian to English translation but with no articles (*some*, *a*, *the*), is likely to be fine for assimilation, but would need substantial post-editing for dissemination. State-of-the-art MT systems are however usually evaluated —even if manually— (and optimized) with respect to their ability to produce translations that resemble references, regardless of the intended application for the system.

Assimilation is by far the main use of MT in number of words translated. It is either explicitly invoked, for instance, by visiting webpages such as Google Translate, or integrated into browsers and social networks. Raw MT may sometimes be the only feasible option,[1] for instance when dealing with user-generated content or ephemeral material (such as product descriptions in e-commerce).

### 1.2 Evaluation of MT for gisting

A straightforward (but costly) way to evaluate MT for gisting *measures* the performance of target-language readers in a text-mediated task —for instance, a software installation task (Castilho et al., 2014)— by using raw MT and compares it with the performance reached using a professional translation of the text.

However, there may be scenarios without an obvious associated task: news, product and service reviews, or literature. On the other hand, even with a clear associated task, task completion evaluation is also quite expensive. It is therefore desirable to have alternative objective indicators which work as good surrogates for actual task-oriented

[1] Twenty-five years ago, (Sager, 1993, p. 261) already hinted at MT-only scenarios: "there may, indeed, be no single situation in which either human or machine would be equally suitable."

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 192–203
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64020

success.

Some authors have proposed eye-tracking (Doherty and O'Brien, 2009; Doherty et al., 2010; Stymne et al., 2012; Doherty and O'Brien, 2014; Castilho et al., 2014; Klerke et al., 2015; Castilho and O'Brien, 2016; Sajjad et al., 2016) as a measure of machine translation usefulness, but the technique is expensive and the evidence gathered is rather indirect and does not have a straightforward interpretation in terms of usefulness.

There are many methods in which informants are asked to *judge* the *quality* of machine-translated sentences, usually as regards their monolingual *fluency* (nativeness, grammaticality), their bilingual *adequacy* (how much of the information in the source sentence is present in the machine-translated sentence), or even monolingual adequacy (how much of the information in the reference sentence is present in the machine-translated sentence); informants may be asked either to *directly assess* MT outputs by giving values to these indicators in a predetermined scale or to *rank* a number of MT outputs for the same source sentence (sometimes being asked to consider aspects such as adequacy, fluency, or both). Direct assessments of adequacy and MT ranking are the official evaluation procedure for the most recent WMT translation shared task campaigns (Bojar et al., 2016, 2017). Other researchers use post-task questionnaires (Stymne et al., 2012; Doherty and O'Brien, 2014; Klerke et al., 2015; Castilho and O'Brien, 2016) to assess the perceived usefulness of MT output.

Direct assessment, ranking or post-task questionnaire evaluation methods are clearly subjective and require informants to make “in vitro” *judgements* about the *quality* of MT outputs, without considering their *usefulness* for a specific “in vivo”, real-world application.

### 1.3 Reading comprehension questionnaires

Reading comprehension questionnaires (RCQ), as used in the assessment of foreign-language learning, are the standard approach to evaluate MT for gisting that measures reader performance in response to MT. Readers answer questions using either a machine-translated or a professionally-translated version of the source text and their performance on the tests (i.e. to what extent they answer questions correctly) using the two sets of texts is then compared. RCQ are however quite costly: a human translation is needed for a control group and questions need to be professionally written and often manually marked.

RCQ has a long history as an MT evaluation method. Tomita et al. (1993), Fuji (1999), and Fuji et al. (2001) evaluate the *informativeness* or *usefulness* of English–Japanese MT by using standardized English-as-a-foreign-language RCQs (TOEFL, TOEIC) which have been machine translated into Japanese and they are sometimes capable of distinguishing MT systems. Jones et al. (2005b), Jones et al. (2005a), Jones et al. (2007), and Jones et al. (2009) use the structure of standardized language proficiency tests (Defence Language Proficiency Test, Interagency Language Roundtable) to evaluate the readability of Arabic–English MT texts. MT'ed documents are found to be harder to understand than professional translations, and that they may be assigned an intermediate level of English proficiency. Berka et al. (2011) collected a set of English short paragraphs in various domains, created yes/no questions in Czech about them, and machine translated the English paragraphs into Czech with different MT systems. They found that outputs produced by different MT systems lead to different accuracy in the annotators' answers. Weiss and Ahrenberg (2012) evaluate comprehension of Polish–English translations using RCQ tests and found that a text with more MT errors have less correct answers than a text with fewer MT errors. Finally, Stymne et al. (2012) use RCQ to validate eye-tracking as a tool for MT error analysis for English–Swedish. Interestingly, for one of their systems, the number of correct answers in the RCQ tests were higher than for the human translation. However, test takers were more confident in answering questions about the human translations than about the MT outputs.

In this paper we explore RCQ as a measure of MT quality by using the CREG-mt-eval corpus (Scarton and Specia, 2016). In contrast to previous work, this paper presents an evaluation of MT quality based on open questions that have different levels of difficulty (as presented in Section 2) for a considerable amount of documents (36 in contrast to only 2 analysed by Weiss and Ahrenberg (2012)).

### 1.4 An alternative: evaluation via gap-filling

An alternative approach to RCQs, *gap filling* (GF), has been recently proposed (Trosterud and Unhammer, 2012; O'Regan and Forcada, 2013; Ageeva et al., 2015; Jordan-Núñez et al., 2017) based on another typical way of measuring reading comprehension: *cloze* (or *closure*) testing (Taylor, 1953). Instead of a question, readers get an incomplete sentence with one or more words replaced by gaps, and are asked to fill the gaps. Indeed, GF may be seen as equivalent to the answering of simple reading comprehension questions: for instance, a question like *Who was the president of the Green Party in 2011?* would be equivalent to the sentence with one gap *In 2011, _________ was the president of the Green Party.*

GF tasks are prepared by automatically punching gaps in *reference* sentences taken from a professional translation of the source text. Informants are given the machine-translated sentence as a "hint" for the gap-filling task; therefore, we may view GF as a way of automatically generating questions to evaluate the MT output. The evaluation measure is the proportion of gaps that can be successfully filled using MT as a hint. This can be compared with the success rate in the case where no hint (MT) is provided, to give an estimate of the usefulness of MT output.

Note that *cloze* testing evaluation of machine translation was attempted decades ago in a completely different *readability* setting: gaps were then punched *in machine-translated output* and informants tried to complete them without any further hint (Crook and Bishop, 1965; Sinaiko and Klare, 1972). This work was reviewed and extended later by Somers and Wild (2000). But filling gaps in machine-translated output may be unnecessarily challenging and therefore make evaluation less adequate: for instance, informants would sometimes have to fill gaps in disfluent or ungrammatical text, which is much harder than filling them in a fluent, professionally translated reference, or, even in fluent output, a crucial content word that has been removed may be very hard to guess unless the surrounding text is very redundant. Moreover, the GF method described here has an easier interpretation in terms of its analogy to RCQ.

This paper systematically builds upon previous work on GF to obtain experimental evidence that gap-filling is a viable, lower-cost alternative to RCQ evaluation. Its main **contributions** are:

- While Trosterud and Unhammer (2012), O'Regan and Forcada (2013), and Ageeva et al. (2015) used GF just to demonstrate the usefulness of a single rule-based MT system for each language pair studied, this paper, like Jordan et al.'s (2017), performs a comparison of several MT systems for the same language pair.

- Previous work (Trosterud and Unhammer, 2012; O'Regan and Forcada, 2013; Ageeva et al., 2015; Jordan-Núñez et al., 2017) simply assumes the validity of GF as an evaluation method for MT gisting, in some cases arguing about its equivalence to RCQ. Ours is the first work to actually compare GF and RCQ evaluation of the same MT systems.

- Previous work used sentences (Trosterud and Unhammer, 2012; O'Regan and Forcada, 2013; Ageeva et al., 2015) or short excerpts of text (Jordan-Núñez et al., 2017), but did not study the influence of a larger, document-level machine-translated context around the target sentence, as it is done here.

- This paper explores for the first time a gap-positioning strategy based on an approximate computation of gap entropy, and compares it to random placing of gaps.

The paper is organized as follows: section 2 describes the design and implementation of both evaluation methods, RCQ and GF; then section 3 reports and discusses the results obtained; and, finally, concluding remarks (section 4) close the paper.

## 2 Methodology

### 2.1 Data and informants

We use an extended version of CREG-mt-eval (Scarton and Specia, 2016), a version of the expert-built CREG reading comprehension corpus (Ott et al., 2012) for 2nd-language learners of German. CREG was originally created to build and evaluate systems that automatically correct answers to open questions. CREG-mt-eval contains 108 source (German) documents with different domains, including literature, news, job adverts, and others (on average 372 words and 33 sentences per document). The original documents were machine-translated in December 2015 into English using four systems: an in-house baseline[2] statistical phrase-based Moses (Koehn et al., 2007) system trained on WMT 2015 data (Bojar et al., 2015), Google Translate,[3] Bing[4] and Systran.[5] CREG-mt-eval also contains professional translations of a subset of 36 documents (90–1500 words) as a control group to check whether the questions are adequate for the task. All questions from the CREG original questionnaires (in German) were professionally translated to English. On average, there are 8.8 questions per document.

The questions in CREG-mt-eval are classified (Meurers et al., 2011) as: *literal*, when they can be answered directly from the text and refer to explicit knowledge, such as names, dates (79% of the total number of questions); *reorganization*, also based on literal text understanding, but requiring the combination of information from different parts of the text (12% of the total number of questions); and *inference*, which involve combining literal information with world knowledge (9% of the total number of questions).

Following Scarton and Specia (2016), test takers (informants) for both GF and RCQ were fluent English-speaking volunteers, staff and students at the University of Sheffield, who were paid (with a 10 GBP online gift certificate) to complete the task.

### 2.2 Reading comprehension questionnaire task

For the version of CREG-mt-eval used herein, thirty informants were given a set of six documents each and answered three to five questions per document, using only the English document (either machine- or human-translated) provided. Therefore, for each of the 36 original documents, questions were answered using each machine translation system or the human translation. Each document was only evaluated by one informant. The original German document was not given. The guidelines were similar to those used in other reading comprehension tests: test takers were asked to answer the questions based on the document provided. They were also advised to read the questions first and then look for the information required on the text in order to speed up the task. Questions in CREG-mt-eval were marked as proposed by Ott et al. (2012): *correct answer* (1 mark), if the answer is correct and complete; *extra concept* (0.75 marks), when incorrect additional concepts are added; *missing concept* (0.5 marks), when important concepts are missing; *blend* (0.25 marks) when there are both extra and missing concepts; and *incorrect* (0 marks), when the answer is incorrect or missing.

Given the marks and the type of question, RCQ overall scores ($f$) are calculated as:

$$f = \alpha \cdot \frac{1}{N_l} \sum_{k=1}^{N_l} l_k + \beta \cdot \frac{1}{N_r} \sum_{k=1}^{N_r} r_k + \gamma \cdot \frac{1}{N_i} \sum_{k=1}^{N_i} i_k,$$

where $N_l$, $N_r$ and $N_i$ are the number of literal, reorganization and inference questions, respectively, $l_k$, $r_k$ and $i_k$ are real values between 0 and 1, according to the mark of question $k$, and $\alpha$, $\beta$ and $\gamma$ are weights for the different types of questions.

We experiment with three different types of scores: *simple* (same weight for all question types: $\alpha = \beta = \gamma = 1.0$), i.e. marks are averaged giving all questions the same importance; *weighted*, i.e. marks are averaged using different weights for different types of question ($\alpha = 1$, $\beta = 2$ and $\gamma = 3$);[6] and *literal*, where only marks for literal questions are used to compute the average quality score ($\alpha = 1$, $\beta = \gamma = 0$). The last score is interesting because literal questions are the most similar to gap-filling problems and correspond to almost 80% of the corpus and they should be easier to answer than other types. Therefore, problems in answering a literal question may be a sign of a bad quality translation.

Figure 1 shows an example of the questionnaires presented to the test takers. In this example, the first, second and last questions are inference questions, whilst the third and fourth questions are literal questions.

### 2.3 Gap filling task

Twenty different kinds of configurations were used in problems posed to informants. Sixteen configurations used the four MT systems to generate hints, in two modalities (showing the full

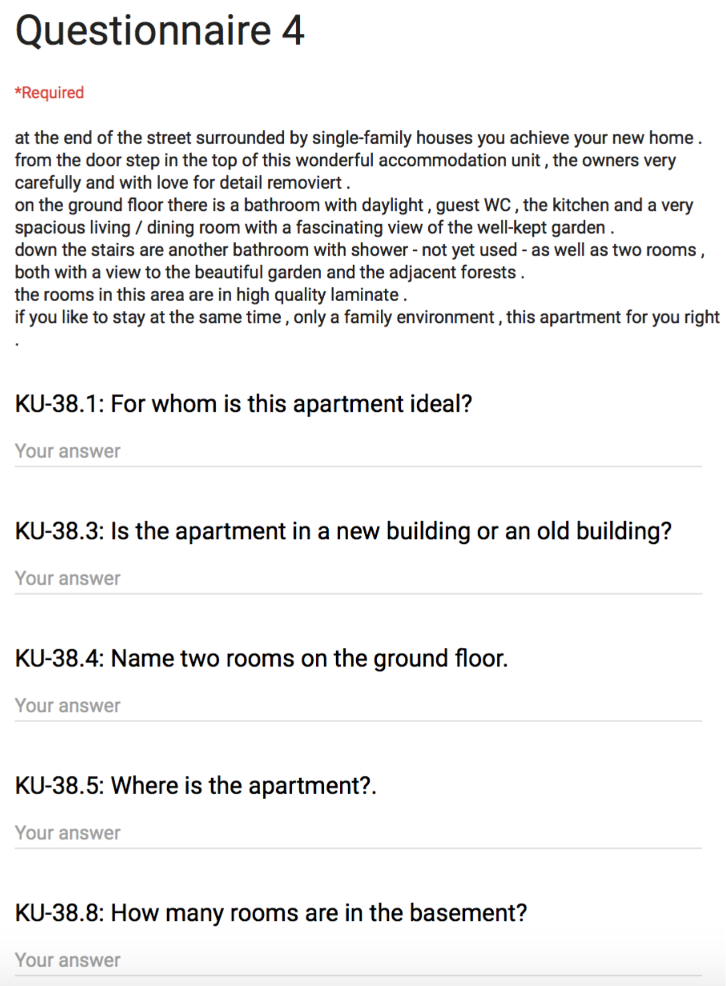
[2] http://www.statmt.org/moses/?n=moses.baseline

[3] http://translate.google.co.uk/, presumably a statistical system at that time.

[4] https://www.bing.com/translator/, also presumably a statistical system at that time.

[5] http://www.systransoft.com/, presumably a hybrid rule-based / statistical system at that time.

[6] These values reflect the expected relative difficulty of questions: inference harder than reorganization, and reorganization harder than literal.



Questionnaire 4

*Required

at the end of the street surrounded by single-family houses you achieve your new home .
from the door step in the top of this wonderful accommodation unit , the owners very carefully and with love for detail removiert .
on the ground floor there is a bathroom with daylight , guest WC , the kitchen and a very spacious living / dining room with a fascinating view of the well-kept garden .
down the stairs are another bathroom with shower - not yet used - as well as two rooms , both with a view to the beautiful garden and the adjacent forests .
the rooms in this area are in high quality laminate .
if you like to stay at the same time , only a family environment , this apartment for you right
.

KU-38.1: For whom is this apartment ideal?

Your answer

KU-38.3: Is the apartment in a new building or an old building?

Your answer

KU-38.4: Name two rooms on the ground floor.

Your answer

KU-38.5: Where is the apartment?.

Your answer

KU-38.8: How many rooms are in the basement?

Your answer

**Figure 1:** A screenshot of a RCQ questionnaire.

machine-translated document, or just the problem sentence) and with two different gap densities (10% or 20%). We added 4 additional configurations with no hint, using the same two gap densities, and with two different gap-selection strategies (statistical language model entropy and random).

The gap entropy at position $k$ of sentence $w_1^N$ is given by,

$$H(k, w_1^N) = -\sum_{x \in V} p(x|w_1^n, k) \log_2 p(x|w_1^n, k),$$

with $V$ the target vocabulary (including the unknown word UNK), and with

$$p(x|w_1^n, k) = \frac{p(w_1^{k-1} x w_{k+1}^N)}{\sum_{x' \in V} p(w_1^{k-1} x' w_{k+1}^N)},$$

estimated using a 3-gram language model trained trained using KenLM (Heafield, 2011) on the English NewsCommentary version 8 corpus.[7] Gaps are punched in order of decreasing entropy, disallowing gaps at stop-words or punctuation, and ensuring that two gaps are never consecutive or separated only by stop-words or punctuation.

To select important sentences for the test, for each of the reference documents, the best single-sentence summary was selected as the problem sentence using GenSim.[8]

Each of 60 informants was given exactly one problem per document. Problem configurations were assigned such that each informant tackled at least one problem in each configuration, and each document was evaluated 3 times in each configuration. The mean time per problem was about 1 minute.

To create the user interface for the task we modified[9] Ageeva et al.'s (2015) version of an older

[7] http://www.statmt.org/wmt13

[8] https://rare-technologies.com/text-summarization-with-gensim/; the percentage of text to be kept in the summary is reduced until it contains a single sentence.

[9] https://github.com/mlforcada/Appraise

version (2014) of Federmann's (2012) Appraise.[10] Each problem was presented in Appraise in a single screen, divided in three sections. The top of each screen reminded informants about the objective of the task. Immediately below, a machine-translated *Hint text* is provided for those 16 configurations that have one. The sentence in the hint text corresponding to the problem sentence is highlighted when a complete document is provided. At the bottom of the screen, the *Problem sentence* containing the gaps to be filled is provided. Figure 2 shows a screenshot of the interface, where a whole machine-translated document is shown as a hint, with the key sentence highlighted. The score for each problem and configuration is simply the ratio of correctly filled gaps.

## 3 Results

Table 1 shows, for each system, the averaged informant performance (see Appendix A for details) for the GF and RCQ quality scores explained previously; BLEU and NIST scores are also given as a reference. In view that score distributions are actually very far from normality, the usual significance tests (such as Welch's $t$-test) are not applicable; therefore, statistical significances of differences between RCQ and GF scores will be reported throughout using the distribution-agnostic Kolmogorov–Smirnov test.[11] Note that previous work in RCQ did not provide statistical significance when comparing different hinting conditions, and that only Jordan et al. (2017) provided that information for GF.

### 3.1 Reading comprehension questionnaire scores

According to all three variations of RCQ scores, and contrary to BLEU and NIST, Systran appears to be better than the homebrew Moses. The RCQ scores for the professionally translated documents ('Human' row on the table) are higher than those for the best MT system, which shows that the questions are answerable from the texts and that informants did follow the guidelines as expected.

We also report the statistical significance of score differences and find (a) the only statistically significant difference at $\alpha < 0.05$ between MT systems for any score type is between Google and the homebrew Moses; (b) all three scores of Bing, Google and Systran are statistically indistinguishable among them; (c) some (but not all) scores obtained with the professional translation are not statistically different from those obtained with Google, Bing or Systran MT output; and (d) all three scores obtained with the professional translation are statistically distinguishable from those with Moses output.

### 3.2 Gap-filling

**Gap placement strategy:** Filling of gaps in the absence of a hint was done in two configurations: one where gaps were punched at random, and one where gaps were punched where LM entropy was maximum. Entropy appears to make gap filling more difficult in the absence of hints (19.6% vs. 25.8% success rate) The value of $p_{\mathrm{KS}} = 0.081$, above the customary $\alpha = 0.05$ significance threshold, would however tentatively support our use of entropy-selected gaps in all situations where MT was used as a hint.

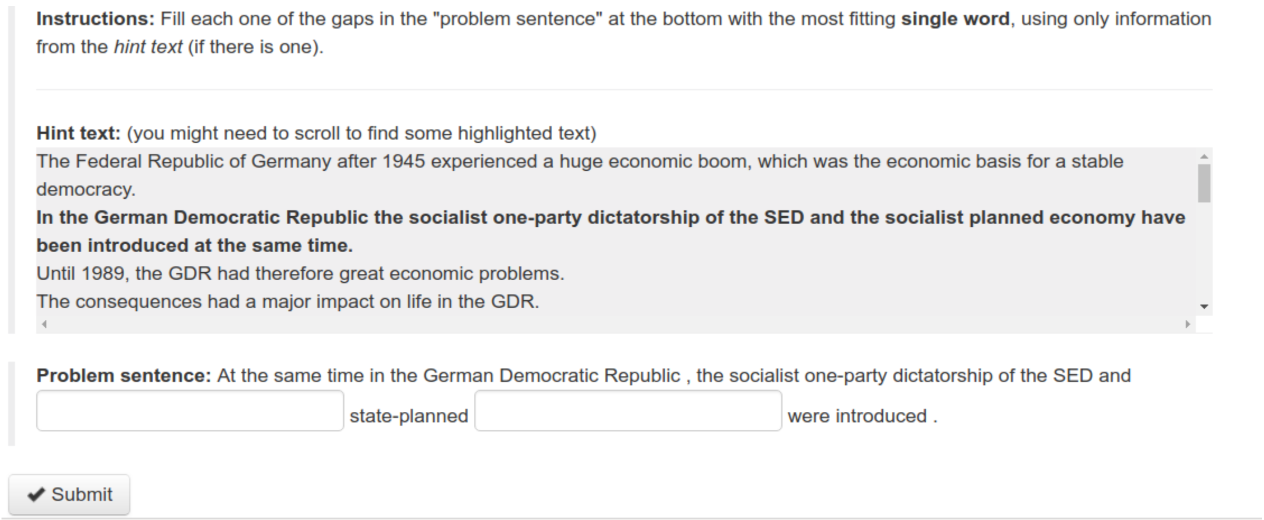
**Comparing MT systems:** Taking all MT systems together, one can see that the success rate (58%) is, as expected, 3 times larger than that obtained without MT using the entropy-driven gap placing strategy (19%) and this difference is statistically significant. The homebrew Moses system is the least helpful (55.9%), and Bing the most helpful (62.6%), but the only statistically significant difference is between these two ($p_{\mathrm{KS}} = 0.005$) and between Bing and Systran ($p_{\mathrm{KS}} = 0.044$). Even with 432 problems solved for each system, MT systems were hard to distinguish by success rate (Jordan et al. (2017) report clearer differences between systems, but the paper does not clarify whether they are running the same problems through all MT systems to ensure the independence of their comparisons).

Figure 3 shows box-and-whisker plots of the distribution of performance across all 60 informants for each MT system. The large overlap observed among the four MT systems illustrates how hard it is to simply average gap-filling scores to evaluate them.

Even if annotators are quite different, each one of them may still be consistent in the relative scores they give to different MT systems. Plotting the average score each informant gives to each MT system against their average score for all systems after removing four clearly outlying in-

[10] https://github.com/cfedermann/Appraise

[11] https://en.wikipedia.org/wiki/Kolmogorov-Smirnov_test



**Instructions:** Fill each one of the gaps in the "problem sentence" at the bottom with the most fitting **single word**, using only information from the *hint text* (if there is one).

**Hint text:** (you might need to scroll to find some highlighted text)

The Federal Republic of Germany after 1945 experienced a huge economic boom, which was the economic basis for a stable democracy.
**In the German Democratic Republic the socialist one-party dictatorship of the SED and the socialist planned economy have been introduced at the same time.**
Until 1989, the GDR had therefore great economic problems.
The consequences had a major impact on life in the GDR.

**Problem sentence:** At the same time in the German Democratic Republic , the socialist one-party dictatorship of the SED and ____ state-planned ____ were introduced .

**Figure 2:** A screenshot of the gap-filling evaluation interface, showing a whole machine-translated document as a hint (with the key sentence highlighted).

| | BLEU | NIST | RCQ scores | | | GF scores | | |
|---|---|---|---|---|---|---|---|---|
| | | | Simple | Weighted | Literal | Overall | 10% | 20% |
| Google | **0.306** | **4.66** | **0.753** | **0.748** | 0.776 | 0.592 | 0.565 | 0.619 |
| Bing | 0.281 | 4.40 | 0.709 | 0.695 | 0.734 | **0.618** | **0.595** | **0.640** |
| Homebrew | 0.241 | 4.51 | 0.594 | 0.577 | 0.608 | 0.550 | 0.547 | 0.553 |
| Systran | 0.203 | 3.05 | 0.680 | 0.670 | 0.701 | 0.569 | 0.544 | 0.595 |
| MT Average | | | 0.684 | 0.673 | 0.705 | 0.582 | 0.563 | 0.602 |
| Human | 1.000 | 10.0 | 0.813 | 0.810 | 0.872 | | | |
| No hint (random) | | | | | | 0.258 | 0.302 | 0.213 |
| No hint (entropy) | | | | | | 0.193 | 0.191 | 0.195 |
| No hint (average) | | | | | | 0.225 | 0.247 | 0.204 |

**Table 1:** A comparison of BLEU and NIST scores, RCQ marks in the three possible weightings, and GF success rates at different densities.

formants, Pearson correlations are only moderate (ranging between 0.47 and 0.73), and the slopes $a_{\text{system}}$ of line fits of the form $\text{score(system)} = a_{\text{system}}\text{score(all)}$ show the same ranking as average scores: $a_{\text{homebrew}} = 0.95$, $a_{\text{Systran}} = 0.97$, $a_{\text{Google}} = 1.00$, $a_{\text{Bing}} = 1.06$, but are very close to each other and their confidence intervals overlap substantially.

**Effect of context:** In half of the configurations with MT hints, a single machine-translated sentence was shown; in the other half, the whole machine-translated document was shown as a hint. The results indicate that extended context, instead of helping, seems to make the task slightly more difficult (58.3% vs. 59.5% success rate), but differences are not statistically significant; therefore, GF scores in Table 1 are average scores obtained with and without context. This supports evaluation through simpler GF tasks based on single-sentence hints.

**Effect of gap density:** Gaps were punched with two different densities, 10% and 20%, to check if a higher gap density would make the problem harder. Contrary to intuition, the task becomes easier when gap density is higher, and the result is statistically significant ($p_{\text{KS}} < 0.001$). This unexpected result is however easily explained as follows: problems with 20% gap density contain all of the high-entropy gaps present in 10% problems, plus additional lower-entropy gaps, which are easier to fill successfully, and therefore, the average success rate rises. In the no-hint situation, however, as shown in Table 1, higher densities would seem to make the problem harder, perhaps because the only information available to fill the gaps comes from the problem sentence itself,

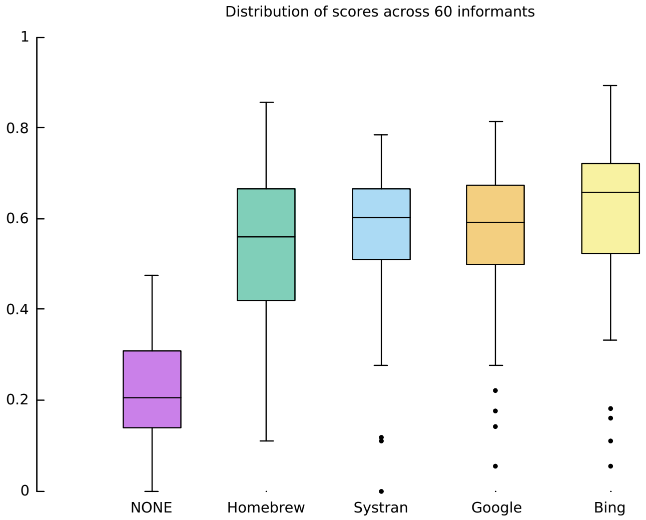


**Figure 3:** Box-and-whisker plots of the distribution of informant performance for each MT system.

and higher gap densities substantially reduce the number of available content words in the sentence. However, the differences are not statistically significant.

**Gap density and MT evaluation:** When comparing MT systems using only the 10% gap density problems, no differences are found to be statistically significant. This means that for very hard gaps, systems would appear to behave similarly. When selecting a value of 20% for the gap density (some easier gaps are included), Bing and Google do appear to be significantly better than the homebrew Moses.

**Inter-annotator agreement:** As 3 different informants filled the gaps for exactly the same set of problems and configurations, with 20 such sets available, we studied the pairwise Pearson correlation $r$ of their GF success in each of the 36 problems.[12] All values of $r$ were found to be positive, averaging around 0.58, a sign of rather good inter-annotator agreement. After removing two outlying informants ($r < 0.1$), results did not appreciably change.

**Allowing for synonyms:** The GF success scores reported thus far have been computed by giving credit only to *exact* matches. We have studied giving credit to *synonyms* observed in informant work, namely to those appearing at least twice (in the work of all informants) that, according to one of the authors, preserved the meaning of the problem sentence, or were trivial spelling or case variations. A total of 124 frequent valid substitutions were considered. As expected, GF success rates (see table 2) increase considerably, for example, from 22.7% to 32.2% for no hint, or from 58.9% to 75.5% for all systems averaged. The relative ranking of MT systems is maintained; the statistical significance of the homebrew Moses results versus Bing results is maintained, and two additional statistically significant differences appear: Google vs. homebrew Moses and Systran vs. homebrew Moses. The statistical significance of the effect of gap density disappears when allowing for synonyms. This indicates that it would be beneficial to assign credit to *synonyms* if the necessary language resources are available or if further analysis of actual GF results is feasible.

### 3.3 Correlation between GF and RCQ

One of our main goals was to explore whether GF would be able to reproduce the results of the established method in the field, RCQ. Table 1 shows reasonable agreement between RCQ and GF scores: both give the homebrew Moses system the worst score, and commercial statistical systems (Bing and Google) get the best scores. Also, as commonly found for subjective *judgements* (for example, Callison-Burch et al. (2006)), BLEU and NIST penalize the rule-based Systran system with respect to the statistical homebrew system, while *measurements* of human performance do not, but the differences observed are however not statistically significant.

[12] The usual Fleiss' kappa statistic cannot be applied here because the labels are not nominal or taken from a discrete set, but rather numerical success rates.

|  | GF scores with synonyms |  |  | GF scores without synonyms |  |  |
|---|---|---|---|---|---|---|
| System | Overall | 10% | 20% | Overall | 10% | 20% |
| Google | 0.757 | 0.711 | 0.776 | 0.592 | 0.565 | 0.619 |
| Bing | **0.795** | **0.785** | **0.804** | **0.618** | **0.595** | **0.640** |
| Homebrew | 0.704 | 0.711 | 0.697 | 0.550 | 0.547 | 0.553 |
| Systran | 0.765 | 0.750 | 0.781 | 0.569 | 0.544 | 0.595 |
| MT Average | 0.755 | 0.746 | 0.765 | 0.582 | 0.563 | 0.602 |
| No hint (random) | 0.339 | 0.379 | 0.299 | 0.258 | 0.302 | 0.213 |
| No hint (entropy) | 0.306 | 0.322 | 0.290 | 0.193 | 0.191 | 0.195 |
| No hint (average) | 0.322 | 0.350 | 0.294 | 0.225 | 0.247 | 0.204 |

**Table 2:** Effect in success rates of allowing for synonyms in GF

On the other hand, GF and RCQ scores assigned to specific (document, MT system) pairs show low correlation. This may be due to the scarcity of RCQ data (only one data point per document–MT system pair, as compared to of 12 data points for GF), or to the fact that, while RCQ takes the whole document into account, GF only looks at a specific sentence. In addition, the RCQ tests and the sentence selected for GF for a given document may not directly correspond, i.e. the information required from the document to answer the RCQ tests may differ from the information required to fill the gaps in a given sentence. This happens because the comprehension questions may target different parts of the text and do not require the sentence selected by our GF approach. A natural follow up of this work is to use sentences for GF directly related to the RCQ tests.

## 4 Concluding remarks

We have compared two methods for the evaluation of MT in gisting applications: the well-established method using reading comprehension questionnaires and an alternative method: gap filling. While RCQ require the manual preparation of questionnaires for each document, and grading of answers to open questions, GF is cheaper, as it only needs reference translations for one or a few sentences in each document and both questions and scores can be obtained automatically. GF is fast and easily crowdsourceable.

In GF, without a hint, we found that entropy-selected gaps appear to be harder than random gaps. We therefore recommend using entropy-selected gaps to discourage guesswork and incentivize annotators to rely on the MT hints. Providing the whole machine-translated document as a hint does not seem to help as compared with providing only the machine-translated version of the problem sentence. This would suggest the possibility of framing GF evaluation around single sentences.

RCQ scores obtained using a machine-translated text range between 70% and 95% of the scores obtained using a professionally-translated text. In GF, the presence of a machine-translated text clearly improves performance (by about 3 times). Both results are a clear indication of the usefulness of raw MT in gisting applications.

Both RCQ and GF rank a low-quality homebrew Moses system worst, but differ as regards the best MT system, although differences are not always statistically significant. It would seem as if informants *make do* with any MT system regardless of small differences in quality. The discriminative power of RCQ and GF evaluations is, however, quite low; this may be due to the scarcity of data; if one expects that the collection of larger amounts of human evaluation data (like the crowdsourced direct assessment (judgement) results described by Bojar et al. (2016)) would increase the discriminative power of the evaluation method, this would be much more feasible using GF, than the more costly RCQ.

**Acknowledgements:** Work supported by the Spanish government through project EFFORTUNE (TIN2015-69632-R) and through grant PRX16/00043 for MLF, and by the European Commission through project Health in my Language (H2020-ICT-2014-1, 644402). CS is supported by the EC project SIMPATICO (H2020-EURO-6-2015, grant number 692819). We would like to thank Dr Ramon Ziai (University of Tübingen) for making the CREG corpus available and answering our questions.

## References

Ekaterina Ageeva, Francis M Tyers, Mikel L Forcada, and Juan Antonio Pérez-Ortiz. 2015. Evaluating machine translation for assimilation via a gap-filling task. In *Proceedings of EAMT*, pages 137–144.

Jan Berka, Martin Černý, and Ondřej Bojar. 2011. Quiz-based evaluation of machine translation. *The Prague Bulletin of Mathematical Linguistics*, 95:77–86.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 131–198, Berlin, Germany.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation. In *Proceedings of the Second Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 169–214, Copenhagen, Denmark.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *EACL*, volume 6, pages 249–256.

Sheila Castilho and Sharon O'Brien. 2016. Evaluating the Impact of Light Post-Editing on Usability. In *The Tenth International Conference on Language Resources and Evaluation*, pages 310–316, Portorož, Slovenia.

Sheila Castilho, Sharon O'Brien, Fabio Alves, and Morgan O'Brien. 2014. Does post-editing increase usability? a study with Brazilian Portuguese as target language. In *Proceedings of the 17th Annual conference of the European Association for Machine translation, EAMT 2014*, pages 183–190. European Association for Machine Translation.

M Crook and H Bishop. 1965. Evaluation of machine translation, final report. Technical report, Institute for Psychological Research, Tufts University, Medford, MA.

Stephen Doherty and Sharon O'Brien. 2009. Can MT output be evaluated through eye tracking? In *The 12th Machine Translation Summit*, pages 214–221, Ottawa, Canada.

Stephen Doherty and Sharon O'Brien. 2014. Assessing the Usability of Raw Machine Translated Output: A User-Centred Study using Eye Tracking. *International Journal of Human-Computer Interaction*, 30(1):40–51.

Stephen Doherty, Sharon O'Brien, and Michael Carl. 2010. Eye tracking as an automatic MT evaluation technique. *Machine Translation*, 24:1–13.

Christian Federmann. 2012. Appraise: An open-source toolkit for manual evaluation of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

M. Fuji, N. Hatanaka, E. Ito, S. Kamei, H. Kumai, T. Sukehiro, T. Yoshimi, and H. Isahara. 2001. Evaluation Method for Determining Groups of Users Who Find MT "Useful". In *The Eightth Machine Translation Summit*, pages 103–108, Santiago de Compostela, Spain.

Masaru Fuji. 1999. Evaluation experiment for reading comprehension of machine translation outputs. In *The Seventh Machine Translation Summit*, pages 285–289, Singapore, Singapore.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Douglas Jones, Edward Gibson, Wade Shen, Neil Granoien, Martha Herzog, Douglas Reynolds, and Clifford Weinstein. 2005a. Measuring human readability of machine generated text: three case studies in speech recognition and machine translation. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v:1009–v:1012. IEEE.

Douglas Jones, Martha Herzog, Hussny Ibrahim, Arvind Jairam, Wade Shen, Edward Gibson, and Michael Emonts. 2007. ILR-based MT comprehension test with multi-level questions. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 77–80. Association for Computational Linguistics.

Douglas Jones, Wade Shen, and Martha Herzog. 2009. Machine translation for government applications. *Lincoln Laboratory Journal*, 18(1).

Douglas A. Jones, Edward Gibson, Wade Shen, Neil Granoien, Martha Herzog, Douglas Reynolds, and Clifford Weinstein. 2005b. Measuring Translation Quality by Testing English Speakers with a New Defense Language Proficiency Test for Arabic. In *The*

*International Conference on Intelligence Analysis*, McLean, VA.

Kenneth Jordan-Núñez, Mikel L. Forcada, and Esteve Clua. 2017. Usefulness of MT output for comprehension — an analysis from the point of view of linguistic intercomprehension. In *Proceedings of MT Summit XVI*, volume 1. Research Track, pages 241–253.

Sigrid Klerke, Sheila Castilho, Maria Barrett, and Anders Søgaard. 2015. Reading metrics for estimating task efficiency with MT output. In *The Sixth Workshop on Cognitive Aspects of Computational Language Learning*, pages 6–13, Lisbon, Portugal.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *The Annual Meeting of the Association for Computational Linguistics, demonstration session*, Prague, Czech Republic.

Ramon Ziai Meurers, Niels Ott, and Janina Kopp. 2011. Evaluating Answers to Reading Comprehension Questions in Context: Results for German and the Role of Information Structure. In *TextInfer 2011 Workshop on Textual Entailment*, pages 1–9, Edinburgh, UK.

Jim O'Regan and Mikel L. Forcada. 2013. Peeking through the language barrier: the development of a free/open-source gisting system for Basque to English based on apertium.org. *Procesamiento del Lenguaje Natural*, pages 15–22.

Niels Ott, Ramon Ziai, and Detmar Meurers. 2012. Creation and analysis of a reading comprehension exercise corpus: Towards evaluating meaning in context. In T. Schmidt and K. Worner, editors, *Multilingual Corpora and Multilingual Corpus Analysis*, Hamburg Studies on Multilingualism (Book 14), pages 47–69. John Benjamins Publishing Company, Amsterdam, The Netherlands.

Juan C. Sager. 1993. *Language engineering and translation: consequences of automation*. Benjamins, Amsterdam.

Hassan Sajjad, Francisco Guzman, Nadir Durrani, Houda Bouamor, Ahmed Abdelali, Irina Teminkova, and Stephan Vogel. 2016. Eyes Don't Lie: Predicting Machine Translation Quality Using Eye Movement. In *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1082–1088, San Diego, CA.

Carolina Scarton and Lucia Specia. 2016. A reading comprehension corpus for machine translation evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

H. Wallace Sinaiko and George R. Klare. 1972. Further experiments in language translation. *International Journal of Applied Linguistics*, 15:1–29.

Harold Somers and Elizabeth Wild. 2000. Evaluating machine translation: the cloze procedure revisited. In *Translating and the Computer 22: Proceedings of the Twenty-second International Conference on Translating and the Computer*.

Sara Stymne, Henrik Danielsson, Sofia Bremin, Hongzhan Hu, Johanna Karlsson, Anna Prytz Lillkull, and Martin Wester. 2012. Eye Tracking as a Tool for Machine Translation Error Analysis. In *The 8th International Conference on Language Resources and Evaluation*, pages 1121–1126, Istanbul, Turkey.

Wilson L Taylor. 1953. "Cloze procedure": a new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433.

Masaru Tomita, Shirai Masako, Junya Tsutsumi, Miki Matsumura, and Yuki Yoshikawa. 1993. Evaluation of MT systems by TOEFL. In *The Fifth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 252–265, Kyoto, Japan.

Trond Trosterud and Kevin Brubeck Unhammer. 2012. Evaluating North Sámi to Norwegian assimilation RBMT. In *Proceedings of the Third International Workshop on Free/Open-Source Rule-Based Machine Translation (FreeRBMT 2012)*.

Sandra Weiss and Lars Ahrenberg. 2012. Error profiling for evaluation of machine-translated text: a Polish–English case study. In *LREC*, pages 1764–1770.

## A Supplemental material

**Raw gap-filling results** for 2159 problems,[13] 60 informants, 36 documents, and 20 configurations, are available for download at the following address: http://www.dlsi.ua.es/~mlf/wmt2018/raw-gap-filling-results.csv.

**Raw reading comprehension test results** for 36 documents, four different MT systems (Google, Bing, Moses and Systran) and one human reference are available, totalling 180 documents. Each document was assessed by one test taker. The markings for questions available in each document and the final document scores

[13] Should have been 2160 = 36 × 60, but data for one specific document, informant and configuration, was lost due to a bug in the Appraise system.

used in this paper (namely simple, weighted or literal) are available for download at: http://www.dlsi.ua.es/~mlf/wmt2018/raw-reading-comprehension-results.csv.

# Simple Fusion: Return of the Language Model

**Felix Stahlberg**[†*] and **James Cross**[‡] and **Veselin Stoyanov**[‡]
[†]Department of Engineering, University of Cambridge, UK
[‡]Applied Machine Learning, Facebook, Menlo Park, CA, USA
fs439@cam.ac.uk, jcross@fb.com, vesko.st@gmail.com

## Abstract

Neural Machine Translation (NMT) typically leverages monolingual data in training through backtranslation. We investigate an alternative simple method to use monolingual data for NMT training: We combine the scores of a pre-trained and fixed language model (LM) with the scores of a translation model (TM) while the TM is trained from scratch. To achieve that, we train the translation model to predict the residual probability of the training data added to the prediction of the LM. This enables the TM to focus its capacity on modeling the source sentence since it can rely on the LM for fluency. We show that our method outperforms previous approaches to integrate LMs into NMT while the architecture is simpler as it does not require gating networks to balance TM and LM. We observe gains of between +0.24 and +2.36 BLEU on all four test sets (English-Turkish, Turkish-English, Estonian-English, Xhosa-English) on top of ensembles without LM. We compare our method with alternative ways to utilize monolingual data such as backtranslation, shallow fusion, and cold fusion.

## 1 Introduction

Machine translation (MT) relies on parallel training data, which is difficult to acquire. In contrast, monolingual data is abundant for most languages and domains. Traditional statistical machine translation (SMT) effectively leverages monolingual data using language models (LMs) (Brants et al., 2007). The combination of LM and TM in SMT can be traced back to the noisy-channel model which applies the Bayes rule to decompose a translation system (Brown et al., 1993):

$$\begin{aligned}\hat{\mathbf{y}} &= \operatorname*{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \\ &= \operatorname*{argmax}_{\mathbf{y}} P_{TM}(\mathbf{x}|\mathbf{y}) P_{LM}(\mathbf{y})\end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_m)$ is the source sentence, $\mathbf{y} = (y_1, \ldots, y_n)$ is the target sentence, and $P_{TM}(\cdot)$ and $P_{LM}(\cdot)$ are translation model and language model probabilities.

In contrast, NMT (Sutskever et al., 2014; Bahdanau et al., 2014) uses a discriminative model and learns the distribution $P(\mathbf{y}|\mathbf{x})$ directly end-to-end. Therefore, the vanilla training regimen for NMT is not amenable to integrating an LM or monoglingual data in a straightforward manner.

An early attempt to use LMs for NMT, also known as *shallow fusion*, combines LM and NMT scores at inference time in a log-linear model (Gulcehre et al., 2015, 2017). In contrast, we integrate the LM scores during NMT training. Our training procedure first trains an LM on a large monolingual corpus. We then hold the LM fixed and train the NMT system to optimize the combined score of LM and NMT on the parallel training set. This allows the NMT model to focus on modeling the source sentence, while the LM handles the generation based on the target-side history. Sriram et al. (2017) explored a similar idea for speech recognition using a gating network for controlling the relative contribution of the LM. We show that our simpler architecture without an explicit control mechanism is effective for machine translation. We observe gains of up to more than 2 BLEU points from adding the LM to TM training. We also show that our method can be combined with backtranslation (Sennrich et al., 2016a), yielding further gains over systems without LM.

[0]This work was done when the first author was on an internship at Facebook.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 204–211
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64021

## 2 Related Work

### 2.1 Inference-time Combination

*Shallow fusion* (Gulcehre et al., 2015) integrates an LM by changing the decoding objective to:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\text{argmax}} \log P_{\text{TM}}(\mathbf{y}|\mathbf{x}) + \lambda \log P_{\text{LM}}(\mathbf{y}). \quad (2)$$

$P_{\text{LM}}(\cdot)$ is produced by an LSTM-based RNN-LM (Mikolov et al., 2010) which has been trained on monolingual target language data. $P_{\text{TM}}(\cdot)$ can be a typical encoder-decoder Seq2Seq model (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015a). $\lambda$ is a hyper-parameter which is tuned on the development set.

### 2.2 Cold Fusion

Shallow fusion combines a fixed TM with a fixed LM at inference time. Sriram et al. (2017) proposed to keep the LM fixed, but train a sequence to sequence (Seq2Seq) NMT model from scratch which includes the LM as a fixed part of the network. They argue that this approach allows the Seq2Seq network to use its model capacity for the conditioning on the source sequence since the language modeling aspect is already covered by the LM. Their *cold fusion* architecture includes a gating network which learns to regulate the contributions of the LM at each time step. They demonstrated superior performance of cold fusion on a speech recognition task.

### 2.3 Other Approaches

Gulcehre et al. (2015, 2017) suggest to combine a pre-trained RNN-LM with a pre-trained NMT system using a controller network that dynamically adjusts the weights between RNN-LM and NMT at each time step (*deep fusion*). Both deep fusion and $n$-best reranking with count-based LMs have been used in WMT evaluation systems (Jean et al., 2015; Wang et al., 2017). An important limitation of these approaches is that LM and TM are trained independently.

A second line of research augments the parallel training data with additional synthetic data from a monolingual corpus in the target language. The source sentences can be generated with a separate translation system (Schwenk, 2008; Sennrich et al., 2016a) (backtranslation), or simply copied over from the target side (Currey et al., 2017). Since data augmentation methods rely on some balance between real and synthetic data (Sennrich et al., 2016a; Currey et al., 2017; Poncelas et al., 2018), they can often only use a small fraction of the available monolingual data. A third class of approaches change the NMT training loss function to incorporate monolingual data. For example, Cheng et al. (2016); Tu et al. (2017) proposed to add autoencoder terms to the training objective which capture how well a sentence can be reconstructed from its translated representation. However, training with respect to the new loss is often computationally intensive and requires approximations. Alternatively, multi-task learning has been used to incorporate source-side (Zhang and Zong, 2016) and target-side (Domhan and Hieber, 2017) monolingual data. Another way of utilizing monolingual data in both source and target language is to warm start Seq2Seq training from pre-trained encoder and decoder networks (Ramachandran et al., 2017; Skorokhodov et al., 2018). We note that pre-training can be used in combination with our approach.

An extreme form of leveraging monolingual training data is unsupervised NMT (Lample et al., 2017; Artetxe et al., 2017) which removes the need for parallel training data entirely. In this work, we assume to have access to some amount of parallel training data, but aim to improve the translation quality even further by using a language model.

## 3 Translation Model Training under Language Model Predictions

In spirit of the cold fusion technique of Sriram et al. (2017) we also keep the LM fixed when training the translation network. However, we greatly simplify the architecture by removing the need for a gating network. We follow the usual left-to-right factorization in NMT:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{n} P(y_t|y_1^{t-1}, \mathbf{x}). \quad (3)$$

Let $S_{\text{TM}}(y_t|y_1^{t-1}, \mathbf{x})$ be the output of the TM projection layer without softmax, i.e., what we would normally call the logits. We investigate two different ways to parameterize $P(y_t|y_1^{t-1}, \mathbf{x})$ using $S_{\text{TM}}(y_t|y_1^{t-1}, \mathbf{x})$ and a fixed and pre-trained language model $P_{\text{LM}}(\cdot)$: PostNorm and PreNorm.

PostNorm This variant is directly inspired by shallow fusion (Eq. 2) as we turn $S_{\text{TM}}(y_t|y_1^{t-1}, \mathbf{x})$

into a probability distribution using a softmax layer, and sum its log-probabilities with the log-probabilities of the LM, i.e. multiply their probabilities:

$$P(y_t|y_1^{t-1},\mathbf{x}) = \text{softmax}(S_{\text{TM}}(y_t|y_1^{t-1},\mathbf{x})) \cdot P_{\text{LM}}(y_t|y_1^{t-1}). \tag{4}$$

PRENORM Another option is to apply normalization after combining the raw $S_{\text{TM}}(y_t|y_1^{t-1},\mathbf{x})$ scores with the LM log-probability:

$$P(y_t|y_1^{t-1},\mathbf{x}) = \text{softmax}\Big(S_{\text{TM}}(y_t|y_1^{t-1},\mathbf{x}) + \log P_{\text{LM}}(y_t|y_1^{t-1})\Big). \tag{5}$$

### 3.1 Theoretical Discussion of POSTNORM and PRENORM

Note that $P(y_t|y_1^{t-1},\mathbf{x})$ might not represent a valid probability distribution under the POSTNORM criterion since, as component-wise product of two distributions, it is not guaranteed to sum to 1. A way to fix this issue would be to combine TM and LM probabilities in the probability space rather than in the log space. However, we have found that probability space combination does not work as well as POSTNORM in our experiments. We can describe $S_{\text{TM}}(y_t|y_1^{t-1},\mathbf{x})$ under POSTNORM informally as the residual probability added to the prediction of the LM.

It is interesting to investigate what signal is actually propagated into $S_{\text{TM}}(y_t|y_1^{t-1},\mathbf{x})$ when training with the PRENORM strategy. We can rewrite $P(y_t|y_1^{t-1},\mathbf{x})$ as:

$$\begin{aligned} P(y_t|y_1^{t-1},\mathbf{x}) &= \frac{P(y_t,y_1^{t-1}|\mathbf{x})}{P(y_1^{t-1}|\mathbf{x})} \\ &= \frac{P(y_t,\mathbf{x}|y_1^{t-1})}{P(\mathbf{x}|y_1^{t-1})} \\ &= \frac{P(\mathbf{x}|y_t,y_1^{t-1})}{P(\mathbf{x}|y_1^{t-1})} P(y_t|y_1^{t-1}). \end{aligned} \tag{6}$$

Alternatively, we can decompose $P(y_t|y_1^{t-1},\mathbf{x})$ as follows using Eq. 5:

$$\begin{aligned} P(y_t|y_1^{t-1},\mathbf{x}) &= \text{softmax}\Big(S_{\text{TM}}(y_t|y_1^{t-1},\mathbf{x}) + \log P_{\text{LM}}(y_t|y_1^{t-1})\Big) \\ &\propto \exp\Big(S_{\text{TM}}(y_t|y_1^{t-1},\mathbf{x}) + \log P_{\text{LM}}(y_t|y_1^{t-1})\Big) \\ &= \exp(S_{\text{TM}}(y_t|y_1^{t-1},\mathbf{x})) \cdot P_{\text{LM}}(y_t|y_1^{t-1}). \end{aligned} \tag{7}$$

Combining Eq. 6 and Eq. 7 leads to:

$$\exp(S_{\text{TM}}(y_t|y_1^{t-1},\mathbf{x})) \propto \frac{P(\mathbf{x}|y_1^t)}{P(\mathbf{x}|y_1^{t-1})} \tag{8}$$

This means that $S_{\text{TM}}(y_t|y_1^{t-1},\mathbf{x})$ under PRENORM is trained to predict how much more likely the *source* sentence becomes when a particular target token $y_t$ is revealed.

| Language pair | # Sentences |
|---|---|
| Turkish-English (WMT) | 207.7K |
| Estonian-English (WMT) | 2,178.0K |
| Xhosa-English (INTERNAL) | 739.2K |

Table 1: Parallel training data.

| Language | # Sentences | LM Perplexity | |
|---|---|---|---|
| | | dev | test |
| English (WMT) | 26.9M | 91.16 | 87.77 |
| Turkish (WMT) | 3.0M | 59.19 | 70.46 |
| English (INTERNAL) | 20.0M | 105.28 | 108.19 |

Table 2: Monolingual training data.

## 4 Experimental Setup

We evaluate our method on a variety of publicly available and proprietary data sets. For our Turkish-English (tr-en), English-Turkish (en-tr), and Estonian-English (et-en) experiments we use all available parallel data from the WMT18 evaluation campaign to train the translation models. Our language models are trained on *News Crawl 2017*. We use *news-test2017* as development ("dev") set and *news-test2018* as test set.

Additionally, we collected our own proprietary corpus of public posts on Facebook. We refer to it as 'INTERNAL' data set. This corpus consists of monolingual English in-domain sentences and parallel data in Xhosa-English. Training set sizes are summarized in Tables 1 and 2.

Our preprocessing consists of lower-casing, tokenization, and subword-segmentation using joint

| **Architecture Hyperparameters** | |
|---|---|
| Source vocab size (BPE) | 16,000 |
| Target vocab size (BPE) | 16,000 |
| Embedding size (all) | 256 |
| Encoder LSTM units | 512 |
| Encoder layers | 2 |
| Decoder LSTM units | 512 |
| Decoder layers | 2 |
| Attention type | dot product |
| **Training Settings** | |
| Optimization | Vanilla SGD |
| Learning rate | 0.5 |
| Batch size | 32 |
| Label smoothing $\epsilon$ | 0.1 |
| Checkpoint averaging | Last 10 |

Table 3: Summary of NMT settings for all models.

byte pair encoding (Sennrich et al., 2016b) with 16K merge operations. On Turkish, we additionally remove diacritics from the text.

On WMT we use lower-cased SacreBLEU[1] (Post, 2018) to be comparable with the literature.[2] On our internal data we report tokenized BLEU scores.

Our Seq2Seq models are encoder-decoder architectures (Sutskever et al., 2014; Bahdanau et al., 2014) with dot-product attention (Luong et al., 2015b) trained with our PyTorch Translate library.[3] Both decoder and encoder consist of two 512-dimensional LSTM layers and 256-dimensional embeddings. The first encoder layer is bidirectional, the second one runs from right to left. Our training and architecture hyperparameters are summarized in Tab. 3. Our LSTM-based LMs have the same size and architecture as the decoder networks, but do not use attention and do not condition on the source sentence. We run beam search with beam size of 6 in all our experiments.

For each setup we train five models using SGD (batch size of 32 sentences) with learning rate decay and label smoothing, and either select the best one (single system) or ensemble the four best models based on dev set BLEU score.

## 5 Results

Tab. 4 compares our methods PRENORM and POSTNORM on the tested language pairs. Shallow fusion (Sec. 2.1) often leads to minor improvements over the baseline for both single systems and ensembles. We also reimplemented the *cold fusion* technique (Sec. 2.2) for comparison. For our machine translation experiments we report mixed results with cold fusion, with performance ranging between 0.33 BLEU gain on Xhosa-English and slight BLEU degradation in most of our Turkish-English experiments.

**English-Turkish (WMT)**

| **Method** | **Single** | | **4-Ensemble** | |
|---|---|---|---|---|
| | **dev** | **test** | **dev** | **test** |
| Baseline (no LM) | 12.23 | 11.56 | 14.17 | 13.35 |
| Shallow fusion | 12.45 | 11.61 | 14.43 | 13.51 |
| Cold fusion | 12.39 | 11.54 | 14.20 | 13.23 |
| **This work**: PRENORM | 12.82 | 11.93 | **14.78** | 13.41 |
| **This work**: POSTNORM | **13.30** | **12.27** | 14.77 | **13.61** |

**Turkish-English (WMT)**

| **Method** | **Single** | | **4-Ensemble** | |
|---|---|---|---|---|
| | **dev** | **test** | **dev** | **test** |
| Baseline (no LM) | 16.14 | 16.60 | 18.01 | 18.67 |
| Shallow fusion | 16.11 | 16.70 | 18.01 | 18.67 |
| Cold fusion | 16.25 | 16.21 | 17.99 | 18.40 |
| **This work**: PRENORM | 15.88 | 16.39 | 17.95 | 18.40 |
| **This work**: POSTNORM | **16.59** | **17.03** | **18.38** | **19.17** |

**Estonian-English (WMT)**

| **Method** | **Single** | | **4-Ensemble** | |
|---|---|---|---|---|
| | **dev** | **test** | **dev** | **test** |
| Baseline (no LM) | 16.02 | 16.57 | 16.83 | 17.91 |
| Shallow fusion | 16.02 | 16.57 | 16.83 | 17.91 |
| Cold fusion | 15.40 | 15.99 | 16.48 | 17.79 |
| **This work**: PRENORM | **16.80** | **17.44** | **17.78** | **19.01** |
| **This work**: POSTNORM | 16.43 | 17.10 | 17.62 | 18.63 |

**Xhosa-English (INTERNAL)**

| **Method** | **Single** | | **4-Ensemble** | |
|---|---|---|---|---|
| | **dev** | **test** | **dev** | **test** |
| Baseline (no LM) | 10.39 | 11.49 | 13.87 | 15.43 |
| Shallow fusion | 10.69 | 11.65 | 14.06 | 15.54 |
| Cold fusion | 10.72 | 11.29 | 13.66 | 15.13 |
| **This work**: PRENORM | 11.06 | 12.13 | 14.50 | 16.07 |
| **This work**: POSTNORM | **12.34** | **13.27** | **15.45** | **17.79** |

Table 4: Comparison of our PRENORM and POSTNORM combination strategies with shallow fusion (Gulcehre et al., 2015) and cold fusion (Sriram et al., 2017) under an RNN-LM.

Both of our methods, PRENORM and POSTNORM yield significant improvements in BLEU across the board. We report more consistent gains with POSTNORM than with PRENORM. All our POSTNORM systems outperform both shallow fusion and cold fusion on all language pairs, yielding test set gains of up to +2.36 BLEU (Xhosa-English ensembles).

## 6 Discussion and Analysis

**Backtranslation** A very popular technique to use monolingual data for NMT is backtranslation (Sennrich et al., 2016a). Backtranslation

[1] SacreBLEU signature for tr-en test-2017: BLEU+c.lc+l.tr-en+#.1+s.exp+t.wmt17+tok.13a+v.1.2.10

[2] For translation into Turkish we evaluate after diacritics removal.

[3] `https://github.com/pytorch/translate`

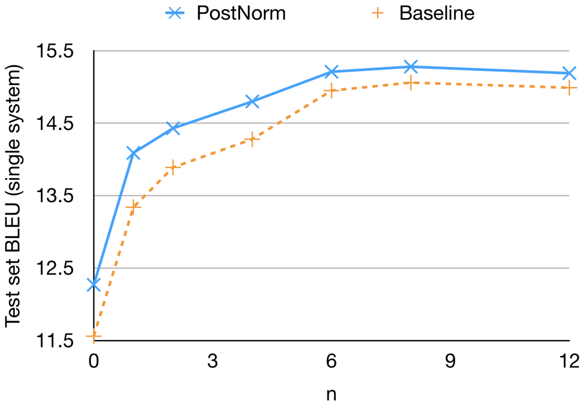


Figure 1: Performance using backtranslation on English-Turkish. Synthetic sentences are mixed at a ratio of 1:$n$ where $n$ is plotted on the $x$-axis.

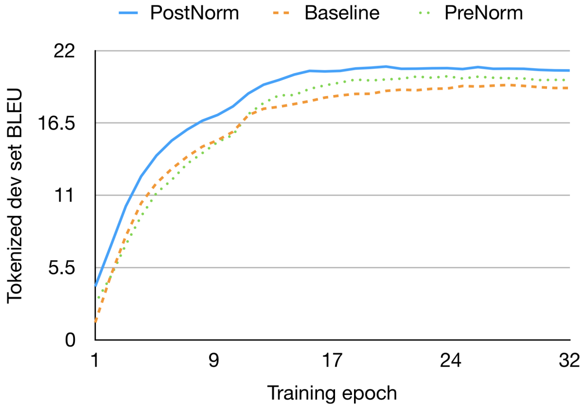


Figure 2: Convergence of NMT training with and without LM on English-Turkish.

**English-Turkish (WMT, single system)**

| Method | Dev set | | Test set | |
|---|---|---|---|---|
| | FFN | RNN | FFN | RNN |
| Baseline (no LM) | 12.23 | | 11.56 | |
| Shallow fusion | 12.25 | 12.45 | 11.53 | 11.61 |
| Cold fusion | 12.33 | 12.39 | 11.51 | 11.54 |
| **This work**: PRENORM | 12.76 | 12.82 | 11.82 | 11.93 |
| **This work**: POSTNORM | 12.65 | 13.30 | 11.79 | 12.27 |

Table 5: Comparison between using a recurrent LM (RNN) and an $n$-gram based feedforward LM (FFN) on English-Turkish.

**English-Turkish (WMT), POSTNORM strategy**

| LM type | | Single | | 4-Ensemble | |
|---|---|---|---|---|---|
| FFN | RNN | dev | test | dev | test |
| | | 12.23 | 11.56 | 14.17 | 13.35 |
| ✓ | | 12.65 | 11.79 | 14.36 | 13.48 |
| | ✓ | 13.30 | 12.27 | 14.77 | 13.61 |
| ✓ | ✓ | 12.86 | 12.02 | 14.72 | 13.70 |

Table 6: Combining an RNN-LM and a feedforward LM with the translation model using the POSTNORM strategy.

uses a reverse NMT system to translate monolingual target language sentences into the source language, and adds the newly generated sentence pairs to the training data. The amount of monolingual data which can be used for backtranslation is usually limited by the size of the parallel corpus as the translation quality suffers when the mixing ratio between synthetic and real source sentences is too large (Poncelas et al., 2018). This is a severe limitation particularly for low-resource MT. Fig. 1 shows that both our baseline system without LM and our POSTNORM system benefit greatly from backtranslation up to a mixing ratio of 1:8, but degrade slightly if this ratio is exceeded. POSTNORM is significantly better than the baseline even when using it in combination with backtranslation.

**Training convergence** We have found that training converges faster under the POSTNORM loss. Fig. 2 plots the training curves of our systems. The baseline (orange curve) reaches its maximum of 19.39 BLEU after 28 training epochs. POSTNORM surpasses this BLEU score already after 12 epochs.

**Language model type** So far we have used recurrent neural network language models (Mikolov et al., 2010, RNN-LM) with LSTM cells in all our experiments. We can also parameterize an $n$-gram language model with a feedforward neural network (Bengio et al., 2003, FFN-LM). In order to compare both language model types we trained a 4-gram feedforward LM with two 512-dimensional hidden layers and 256-dimensional embeddings on Turkish monolingual data. Tab. 5 shows that the PRENORM strategy works particularly well for the $n$-gram LM. However, using an RNN-LM with the POSTNORM strategy still gives the best overall performance. Using both RNN and $n$-gram LM at the same time does not improve translation quality any further (Tab. 6).

**Impact on the TM distribution** With the POSTNORM strategy, the TM still produces a distribution over the target vocabulary as the scores are

| Method | Perplexity | Average entropy |
|---|---|---|
| Baseline (no LM) | 23.46 | 3.19 |
| RNN-LM | 59.19 | 4.66 |
| TM under POSTNORM | 113.69 | 1.82 |

Table 7: Perplexity and average entropies of the distributions generated by our systems on the English-Turkish dev set.

| **Method** | **BLEU** | **Precisions** | | | | **BP** |
|---|---|---|---|---|---|---|
| | | **1-gram** | **2-gram** | **3-gram** | **4-gram** | |
| Baseline (no LM) | 17.91 | 53.0 | 23.7 | 12.3 | 6.6 | 0.996 |
| **This work**: PRENORM | 19.01 | 54.0 | 24.9 | 13.4 | 7.4 | 1.000 |
| Relative improvement | +6.14% | **+1.89%** | **+5.06%** | **+8.94%** | **+12.12%** | – |

Table 8: BLEU $n$-gram precisions for Estonian-English.

| | |
|---|---|
| **Source** | Eestis ja Hispaanias peeti kinni neli Kemerovo grupeeringu liiget |
| **Reference** | Four members of the Kemerovo group arrested in Estonia and Spain |
| **Baseline (no LM)** | In Estonia and Spain, four kemerovo groups were held |
| **This work** (PRENORM) | Four Kemerovo group members were held in Estonia and Spain |
| **Source** | Ta tleb, et elab aastaid hiljem endiselt hirmus. |
| **Reference** | He says that years later, he still lives in fear. |
| **Baseline (no LM)** | He says that, for years, he still lives in fear. |
| **This work** (PRENORM) | He says that many years later he still lives in fear. |
| **Source** | “Ma kardan,” tleb ta. |
| **Reference** | “I’m afraid,” he says. |
| **Baseline (no LM)** | “I fear,” says he. |
| **This work** (PRENORM) | “I am afraid,” he says. |

Table 9: Translation samples from the Estonian-English test set.

normalized before the combination with the LM. This raises a natural question: How different are the distributions generated by a TM trained under POSTNORM loss from the distributions of the baseline system without LM? Tab. 7 gives some insight to that question. As expected, the RNN-LM has higher perplexity than the baseline as it is a weaker model of translation. The RNN-LM also has a higher average entropy which indicates that the LM distributions are smoother than those from the baseline translation model. The TM trained under POSTNORM loss has a much higher perplexity which suggests that it strongly relies on the LM predictions and performs poorly when it is not combined with it. However, the average entropy is much lower (1.82) than both other models, i.e. it produces much sharper distributions.

**Language models improve fluency** A traditional interpretation of the role of an LM in MT is that it is (also) responsible for the fluency of translations (Koehn, 2009). Thus, we would expect more fluent translations from our method than from a system without LM. Tab. 8 breaks down the BLEU score of the baseline and the PRENORM ensembles on Estonian-English into $n$-gram precisions. Most of the BLEU gains can be attributed to the increase in precision of higher order $n$-grams, indicating improvements in fluency. Tab. 9 shows some examples where our PRENORM system produces a more fluent translation than the baseline.

**Training set size** We artificially reduced the size of the English-Turkish training set even further to investigate how well our method performs in low-resource settings (Fig. 3). Our POSTNORM strategy outperforms the baseline regardless of the number of training sentences, but the gains are smaller on very small training sets.

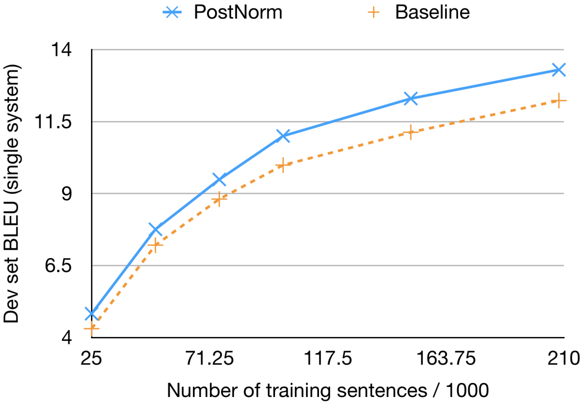


Figure 3: English-Turkish BLEU over training set size.

## 7 Conclusion

We have presented a simple yet very effective method to use language models in NMT which incorporates the LM already into NMT training. We reported significant and consistent gains from using our method in four language directions over two alternative ways to integrate LMs into NMT (*shallow fusion* and *cold fusion*) and showed that our approach works well even in combination with backtranslation and on top of ensembles. Our method leads to faster training convergence and more fluent translations than a baseline system without LM.

## References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, Prague, Czech Republic. Association for Computational Linguistics.

Peter E. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1965–1974. Association for Computational Linguistics.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156. Association for Computational Linguistics.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505. Association for Computational Linguistics.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017. On integrating a language model into neural machine translation. *Computer Speech & Language*, 45:137 – 148.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140. Association for Computational Linguistics.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating backtranslation in neural machine translation. *arXiv preprint arXiv:1804.06189*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. *arXiv preprint arXiv:1804.08771*.

Prajit Ramachandran, Peter Liu, and Quoc Le. 2017. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 383–391. Association for Computational Linguistics.

Holger Schwenk. 2008. Investigations on large-scale lightly-supervised training for statistical machine translation. In *In IWSLT*, pages 182–189.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Ivan Skorokhodov, Anton Rykachevskiy, Dmitry Emelyanenko, Sergey Slotin, and Anton Ponkratov. 2018. Semi-supervised neural machine translation with language models. In *Proceedings of the AMTA 2018 Workshop on Technologies for MT of Low Resource Languages (LoResMT 2018)*, pages 37–44. Association for Machine Translation in the Americas.

Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. 2017. Cold fusion: Training seq2seq models together with language models. *arXiv preprint arXiv:1708.06426*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017. Sogou neural machine translation systems for WMT17. In *Proceedings of the Second Conference on Machine Translation*, pages 410–415. Association for Computational Linguistics.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

# Correcting Length Bias in Neural Machine Translation

**Kenton Murray** and **David Chiang**
Department of Computer Science and Engineering
University of Notre Dame
{kmurray4,dchiang}@nd.edu

## Abstract

We study two problems in neural machine translation (NMT). First, in beam search, whereas a wider beam should in principle help translation, it often hurts NMT. Second, NMT has a tendency to produce translations that are too short. Here, we argue that these problems are closely related and both rooted in label bias. We show that correcting the brevity problem almost eliminates the beam problem; we compare some commonly-used methods for doing this, finding that a simple per-word reward works well; and we introduce a simple and quick way to tune this reward using the perceptron algorithm.

## 1 Introduction

Although highly successful, neural machine translation (NMT) systems continue to be plagued by a number of problems. We focus on two here: the beam problem and the brevity problem.

First, machine translation systems rely on heuristics to search through the intractably large space of possible translations. Most commonly, beam search is used during the decoding process. Traditional statistical machine translation systems often rely on large beams to find good translations. However, in neural machine translation, increasing the beam size has been shown to degrade performance. This is the last of the six challenges identified by Koehn and Knowles (2017).

The second problem, noted by several authors, is that NMT tends to generate translations that are too short. Jean et al. (2015) and Koehn and Knowles address this by dividing translation scores by their length, inspired by work on audio chords (Boulanger-Lewandowski et al., 2013). A similar method is also used by Google's production system (Wu et al., 2016). A third simple method used by various authors (Och and Ney, 2002; He et al., 2016; Neubig, 2016) is a tunable reward added for each output word. Huang et al. (2017) and Yang et al. (2018) propose variations of this reward that enable better guarantees during search.

In this paper, we argue that these two problems are related (as hinted at by Koehn and Knowles) and that both stem from *label bias*, an undesirable property of models that generate sentences word by word instead of all at once.

The typical solution is to introduce a sentence-level correction to the model. We show that making such a correction almost completely eliminates the beam problem. We compare two commonly-used corrections, length normalization and a word reward, and show that the word reward is slightly better.

Finally, instead of tuning the word reward using grid search, we introduce a way to learn it using a perceptron-like tuning method. We show that the optimal value is sensitive both to task and beam size, implying that it is important to tune for every model trained. Fortunately, tuning is a quick post-training step.

## 2 Problem

Current neural machine translation models are examples of locally normalized models, which estimate the probability of generating an output sequence $e = e_{1:m}$ as

$$P(e_{1:m}) = \prod_{i=1}^{m} P(e_i \mid e_{1:i-1}).$$

For any partial output sequence $e_{1:i}$, let us call $P(e' \mid e_{1:i})$, where $e'$ ranges over all possible completions of $e_{1:i}$, the *suffix distribution* of $e_{1:i}$. The suffix distribution must sum to one, so if the model overestimates $P(e_{1:i})$, there is no way for the suffix distribution to downgrade it. This is known as *label bias* (Bottou, 1991; Lafferty et al., 2001).

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 212–223
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64022

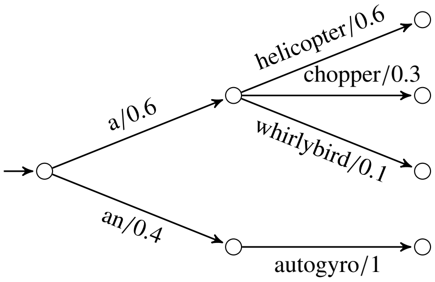


Figure 1: Label bias causes this toy word-by-word translation model to translate French *un hélicoptère* incorrectly to *an autogyro*.

### 2.1 Label bias in sequence labeling

Label bias was originally identified in the context of HMMs and MEMMs for sequence-labeling tasks, where the input sequence $f$ and output sequence $e$ have the same length, and $P(e_{1:i})$ is conditioned only on the partial input sequence $f_{1:i}$. In this case, since $P(e_{1:i})$ has no knowledge of future inputs, it's much more likely to be incorrectly estimated. For example, suppose we had to translate, word-by-word, *un hélicoptère* to *a helicopter* (Figure 1). Given just the partial input *un*, there is no way to know whether to translate it as *a* or *an*. Therefore, the probability for the incorrect translation $P(\text{an})$ will turn out to be an overestimate. As a result, the model will overweight translations beginning with *an*, regardless of the next input word.

This effect is most noticeable when the suffix distribution has low entropy, because even when new input (*hélicoptère*) is revealed, the model will tend to ignore it. For example, suppose that the available translations for *hélicoptère* are *helicopter*, *chopper*, *whirlybird*, and *autogyro*. The partial translation *a* must divide its probability mass among the three translations that start with a consonant, while *an* gives all its probability mass to *autogyro*, causing the incorrect translation *an autogyro* to end up with the highest probability.

In this example, $P(\text{an})$, even though overestimated, is still lower than $P(\text{a})$, and wins only because its suffixes have higher probability. Greedy search would prune the incorrect prefix *an* and yield the correct output. In general, then, we might expect greedy or beam search to alleviate some symptoms of label bias. Namely, a prefix with a low-entropy suffix distribution can be pruned if its probability is, even though overestimated, not among the highest probabilities. Such an observation was made by Zhang and Nivre (2012) in the context of dependency parsing, and we will see next that precisely such a situation affects output length in NMT.

### 2.2 Length bias in NMT

In NMT, unlike the word-by-word translation example in the previous section, each output symbol is conditioned on the entire input sequence. Nevertheless, it's still possible to overestimate or underestimate $p(e_{1:i})$, so the possibility of label bias still exists. We expect that it will be more visible with weaker models, that is, with less training data.

Moreover, in NMT, the output sequence is of variable length, and generation of the output sequence stops when `</s>` is generated. In effect, for any prefix ending with `</s>`, the suffix distribution has zero entropy. This situation parallels example of the previous section closely: if the model overestimates the probability of outputting `</s>`, it may proceed to ignore the rest of the input and generate a truncated translation.

Figure 2 illustrates how this can happen. Although the model can learn not to prefer shorter translations by predicting a low probability for `</s>` early on, at each time step, the score of `</s>` puts a limit on the total remaining score a translation can have; in the figure, the empty translation has score $-10.1$, so that no translation can have score lower than $-10.1$. This lays a heavy burden on the model to correctly guess the total score of the whole translation at the outset.

As in our label-bias example, greedy search would prune the incorrect empty translation. More generally, consider beam search: at time step $t$, only the top $k$ partial or complete translations are retained while the rest are pruned. (Implementations of beam search vary in the details, but this variant is simplest for the sake of argument.) Even if a translation ending at time $t$ scores higher than a longer translation, as long as it does not fall within the top $k$ when compared with partial translations of length $t$ (or complete translations of length at most $t$), it will be pruned and unable to block the longer translation. But if we widen the beam ($k$), then translation accuracy will suffer. We call this problem (which is Koehn and Knowles's sixth challenge) the *beam* problem. Our claim, hinted at by Koehn and Knowles (2017), is that the brevity problem and the beam problem are essentially the same, and that solving one will solve the other.

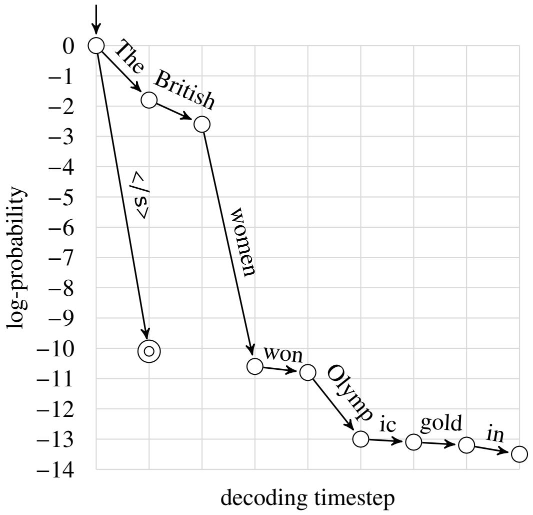


Figure 2: A locally normalized model must determine, at each time step, a "budget" for the total remaining log-probability. In this example sentence, "The British women won Olymp ic gold in p airs row ing," the empty translation has initial position 622 in the beam. Already by the third step of decoding, the correct translation has a lower score than the empty translation. However, using greedy search, a nonempty translation would be returned.

## 3 Correcting Length

To address the brevity problem, many designers of NMT systems add corrections to the model. These corrections are often presented as modifications to the search procedure. But, in our view, the brevity problem is essentially a modeling problem, and these corrections should be seen as modifications to the model (Section 3.1). Furthermore, since the root of the problem is local normalization, our view is that these modifications should be trained as globally-normalized models (Section 3.2).

### 3.1 Models

Without any length correction, the standard model score (higher is better) is:

$$s(e) = \sum_{i=1}^{m} \log P(e_i \mid e_{1:i}).$$

To our knowledge, there are three methods in common use for adjusting the model to favor longer sentences.

*Length normalization* divides the score by $m$ (Koehn and Knowles, 2017; Jean et al., 2015; Boulanger-Lewandowski et al., 2013):

$$s'(e) = s(e) \;/\; m.$$

*Google's NMT system* (Wu et al., 2016) relies on a more complicated correction:

$$s'(e) = s(e) \bigg/ \frac{(5+m)^\alpha}{(5+1)^\alpha}.$$

Finally, some systems add a constant *word reward* (He et al., 2016):

$$s'(e) = s(e) + \gamma m.$$

If $\gamma = 0$, this reduces to the baseline model. The advantage of this simple reward is that it can be computed on partial translations, making it easier to integrate into beam search.

### 3.2 Training

All of the above modifications can be viewed as modifications to the base model so that it is no longer a locally-normalized probability model.

To train this model, in principle, we should use something like the globally-normalized negative log-likelihood:

$$L = -\log \frac{\exp s'(e^*)}{\sum_e \exp s'(e)}$$

where $e^*$ is the reference translation. However, optimizing this is expensive, as it requires performing inference on every training example or heuristic approximations (Andor et al., 2016; Shen et al., 2016).

Alternatively, we can adopt a two-tiered model, familiar from phrase-based translation (Och and Ney, 2002), first training $s$ and then training $s'$ while keeping the parameters of $s$ fixed, possibly on a smaller dataset. A variety of methods, like minimum error rate training (Och, 2003; He et al., 2016), are possible, but keeping with the globally-normalized negative log-likelihood, we obtain, for the constant word reward, the gradient:

$$\frac{\partial L}{\partial \gamma} = -|e^*| + E[|e|].$$

If we approximate the expectation using the mode of the distribution, we get

$$\frac{\partial L}{\partial \gamma} \approx -|e^*| + |\hat{e}|$$

where $\hat{e}$ is the 1-best translation. Then the stochastic gradient descent update is just the familiar perceptron rule:

$$\gamma \leftarrow \gamma + \eta\,(|e^*| - |\hat{e}|),$$

although below, we update on a batch of sentences rather than a single sentence. Since there is only one parameter to train, we can train it on a relatively small dataset.

Length normalization does not have any additional parameters, with the result (in our opinion, strange) that a change is made to the model without any corresponding change to training. We could use gradient-based methods to tune the $\alpha$ in the GNMT correction, but the perceptron approximation turns out to drive $\alpha$ to $\infty$, so a different method would be needed.

## 4 Experiments

We compare the above methods in four settings, a high-resource German–English system, a medium-resource Russian–English system, and two low-resource French–English and English–French systems. For all settings, we show that larger beams lead to large BLEU and METEOR drops if not corrected. We also show that the optimal parameters can depend on the task, language pair, training data size, as well as the beam size. These values can affect performance strongly.

### 4.1 Data and settings

Most of the experimental settings below follow the recommendations of Denkowski and Neubig (2017). Our high-resource, German–English data is from the 2016 WMT shared task (Bojar et al., 2016). We use a bidirectional encoder-decoder model with attention (Bahdanau et al., 2015).[1] Our word representation layer has 512 hidden units, while other hidden layers have 1024 nodes. Our model is trained using Adam with a learning rate of 0.0002. We use 32k byte-pair encoding (BPE) operations learned on the combined source and target training data (Sennrich et al., 2016). We train on minibatches of size 2012 words and validate every 100k sentences, selecting the final model based on development perplexity.

Our medium-resource, Russian–English system uses data from the 2017 WMT translation task, which consists of roughly 1 million training sentences (Bojar et al., 2017). We use the same architecture as our German–English system, but only have 512 nodes in all layers. We use 16k BPE operations and dropout of 0.2. We train on minibatches of 512 words and validate every 50k sentences.

Our low-resource systems use French and English data from the 2010 IWSLT TALK shared task (Paul et al., 2010). We build both French–English and English–French systems. These networks are the same as for the medium Russian-English task, but use only 6k BPE operations. We train on minibatches of 512 words and validate every 30k sentences, restarting Adam when the development perplexity goes up.

To tune our correction parameters, we use 1000 sentences from the German–English development dataset, 1000 sentences from the Russian–English development dataset, and the entire development dataset for French–English (892 sentences)[2]. We initialize the parameter, $\gamma = 0.2$. We use batch gradient descent, which we found to be much more stable than stochastic gradient descent, and use a learning rate of $\eta = 0.2$, clipping gradients for $\gamma$ to 0.5. Training stops if all parameters have an update of less than 0.03 or a max of 25 epochs was reached.

### 4.2 Solving the length problem solves the beam problem

Here, we first show that the beam problem is indeed the brevity problem. We then demonstrate that solving the length problem does solve the beam problem. Tables 1, 2, and 3 show the results of our German–English, Russian–English, and French–English systems respectively. Each table looks at the impact on BLEU, METEOR, and the ratio of the lengths of generated sentences compared to the gold lengths (Papineni et al., 2002; Denkowski and Lavie, 2014). The baseline method is a standard model without any length correction. The reward method is the tuned constant word reward discussed in the previous section. Norm refers to the normalization method, where a hypothesis' score is divided by its length.

#### 4.2.1 Baseline

The top sections of Tables 1, 2, 3 illustrate the brevity and beam problems in the baseline models. As beam size increases, the BLEU and METEOR scores drop significantly. This is due to the brevity problem, which is illustrated by the length ratio numbers that also drop with increased

[1] We use Lamtram (Neubig, 2015) for all experiments and our modifications have been added to the project.

[2] We found through preliminary experiments that this size of dev subset was an adequate trade-off between tuning speed and performance.

| Russian–English (medium) | | Beam Size | | | | | |
|---|---|---|---|---|---|---|---|
| | | 10 | 50 | 75 | 100 | 150 | 1000 |
| baseline | BLEU | 24.9 | 23.8 | 23.6 | 23.3 | 22.5 | 3.7 |
| | METEOR | 30.9 | 30.0 | 29.7 | 29.4 | 28.8 | 12.8 |
| | length | 0.90 | 0.86 | 0.85 | 0.84 | 0.81 | 0.31 |
| reward | BLEU | 26.5 | 26.6 | 26.5 | 26.5 | 26.5 | 25.7 |
| | METEOR | 32.0 | 32.0 | 31.9 | 31.9 | 31.9 | 31.2 |
| | length | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 | 1.02 |
| | $\gamma$ | 0.716 | 0.643 | 0.640 | 0.633 | 0.617 | 0.562 |
| norm | BLEU | 26.2 | 26.3 | 26.3 | 26.3 | 26.3 | 25.3 |
| | METEOR | 31.8 | 31.8 | 31.8 | 31.7 | 31.7 | 31.2 |
| | length | 0.96 | 0.96 | 0.96 | 0.96 | 0.97 | 1.02 |

Table 1: Results of the Russian–English translation system. We report BLEU and METEOR scores, as well as the ratio of the length of generated sentences compared to the correct translations (length). $\gamma$ is the word reward score discovered during training. Here, we examine a much larger beam (1000). The beam problem is more pronounced at this scale, with the baseline system losing over 20 BLEU points when increasing the beam from size 10 to 1000. However, both our tuned length reward score and length normalization recover most of this loss.

| German–English (large) | | Beam Size | | |
|---|---|---|---|---|
| | | 10 | 50 | 75 |
| baseline | BLEU | 29.6 | 28.6 | 28.2 |
| | METEOR | 34.0 | 33.1 | 32.8 |
| | length | 0.95 | 0.90 | 0.89 |
| reward | BLEU | 30.3 | 30.6 | 30.6 |
| | METEOR | 34.9 | 34.8 | 34.9 |
| | length | 1.02 | 1.00 | 1.00 |
| | $\gamma$ | 0.67 | 0.57 | 0.58 |
| norm | BLEU | 30.7 | 31.0 | 30.9 |
| | METEOR | 34.9 | 35.0 | 35.0 |
| | length | 1.00 | 1.00 | 1.00 |

Table 2: Results of the high-resource German–English system. Rows: BLEU, METEOR, length = ratio of output to reference length; $\gamma$ = learned parameter value. While baseline performance decreases with beam size due to the brevity problem, other methods perform more consistently across beam sizes. Length normalization (norm) gets the best BLEU scores, but similar METEOR scores to the word reward.

| French–English (small) | | Beam Size | | | | |
|---|---|---|---|---|---|---|
| | | 10 | 50 | 100 | 150 | 200 |
| baseline | BLEU | 30.0 | 28.9 | 25.4 | 21.9 | 19.4 |
| | METEOR | 32.4 | 31.3 | 28.6 | 25.9 | 24.1 |
| | length | 0.94 | 0.89 | 0.80 | 0.71 | 0.64 |
| reward | BLEU | 29.4 | 29.7 | 29.7 | 29.8 | 29.8 |
| | METEOR | 32.8 | 32.9 | 32.9 | 32.9 | 32.9 |
| | length | 1.03 | 1.03 | 1.03 | 1.03 | 1.03 |
| | $\gamma$ | 1.20 | 1.05 | 1.01 | 0.99 | 0.97 |
| norm | BLEU | 30.7 | 30.8 | 30.7 | 30.7 | 30.7 |
| | METEOR | 32.8 | 32.8 | 32.8 | 32.7 | 32.7 |
| | length | 0.97 | 0.97 | 0.97 | 0.96 | 0.96 |

| English–French (small) | | Beam Size | | | | |
|---|---|---|---|---|---|---|
| | | 10 | 50 | 100 | 150 | 200 |
| baseline | BLEU | 25.8 | 26.1 | 26.1 | 25.5 | 24.3 |
| | METEOR | 47.8 | 47.5 | 47.2 | 46.3 | 44.2 |
| | length | 1.03 | 1.01 | 1.00 | 0.97 | 0.92 |
| reward | BLEU | 25.5 | 25.5 | 25.5 | 25.5 | 25.5 |
| | METEOR | 48.3 | 48.5 | 48.5 | 48.5 | 48.4 |
| | length | 1.05 | 1.05 | 1.05 | 1.05 | 1.05 |
| | $\gamma$ | 0.353 | 0.444 | 0.465 | 0.474 | 0.475 |
| norm | BLEU | 25.4 | 25.5 | 25.5 | 25.5 | 25.5 |
| | METEOR | 48.4 | 48.4 | 48.4 | 48.4 | 48.4 |
| | length | 1.06 | 1.05 | 1.05 | 1.05 | 1.05 |

Table 3: Results of low-resource French–English and English–French systems. Rows: BLEU, METEOR, length = ratio of output to reference length; $\gamma$ = learned parameter value. While baseline performance decreases with beam size due to the brevity problem, other methods perform more consistently across beam sizes. Word reward gets the best scores in both directions on METEOR. Length normalization (norm) gets the best BLEU scores in Fra-Eng due to the slight bias of BLEU towards shorter translations.

| beam | 10 | 50 | 75 | 100 | 150 | 200 |
|---|---|---|---|---|---|---|
| French–English (small) | 6.9 | 27.2 | 52.4 | 71.1 | 105.9 | 176.6 |
| English–French (small) | 12.6 | 44.2 | 67.3 | 88.1 | 107.5 | 111.2 |
| German–English (large) | 6.8 | 132.6 | 1066 | | | |

Table 4: Tuning time on top of baseline training time. Times are in minutes on 1000 dev examples (German–English) or 892 dev examples (French–English). Due to the much larger model size, we only looked at beam sizes up to 75 for German–English.

beam size. For larger beam sizes, the length of the generated output sentences are a fraction of the lengths of the correct translations. For the lower-resource French–English task, the drop is more than 8 BLEU when increasing the beam size from 10 to 150. The issue is even more evident in our Russian-English system where we increase the beam to 1000 and BLEU scores drop by more than 20 points.

#### 4.2.2 Word reward

The results of tuning the word reward, $\gamma$, as described in Section 3.2, is shown in the second section of Tables 1, 2, and 3. In contrast to our baseline systems, our tuned word reward always fixes the brevity problem (length ratios are approximately 1.0), and generally fixes the beam problem. An optimized word reward score always leads to improvements in METEOR scores over any of the best baselines. Across all language pairs, reward and norm have close METEOR scores, though the reward method wins out slightly. BLEU scores for reward and norm also increase over the baseline in most cases, despite BLEU's inherent bias towards shorter sentences. Most notably, whereas the baseline Russian–English system lost more than 20 BLEU points when the beam was increased to 1000, our tuned reward score resulted in a BLEU gain over any baseline beam size. Whereas in our baseline systems, the length ratio decreases with larger beam sizes, our tuned word reward results in length ratios of nearly 1.0 across all language pairs, mitigating many of the issues of the brevity problem.

#### 4.2.3 Wider beam

We note that the beam problem in NMT exists for relatively small beam sizes – especially when compared to traditional beam sizes in SMT systems. On our medium-resource Russian–English system, we investigate the full impact of this problem using a much larger beam size of 1000. In Table 1, we can see that the beam problem is particularly pronounced. The first row of the table shows the uncorrected, baseline score. From a beam of 10 to a beam of 1000, the drop in BLEU scores is over 20 points. This is largely due to the brevity problem discussed earlier. The second row of the table shows the length of the translated outputs compared to the lengths of the correct translations. Though the problem persists even at a beam size of 10, at a beam size of 1000, our baseline system generates less than one third the number of words that are in the correct translations. Furthermore, 37.3% of our translated outputs have sentences of length 0. In other words, the most likely translation is to immediately generate the stop symbol. This is the problem visualized in Figure 2.

However, when we tune our word reward score with a beam of 1000, the problem mostly goes away. Over the uncorrected baseline, we see a 22.0 BLEU point difference for a beam of 1000. Over the uncorrected baseline with a beam of 10, the corrected beam of 1000 gets a BLEU gain of 0.8 BLEU. However, the beam of 1000 still sees a drop of less than 1.0 BLEU over the best corrected version. The word reward method beats the uncorrected baseline and the length normalization correction in almost all cases.

#### 4.2.4 Short sentences

Another way to demonstrate that the beam problem is the same as the brevity problem is to look at the translations generated by baseline systems on shorter sentences. Figure 3 shows the BLEU scores of the Russian–English system for beams of size 10 and 1000 on sentences of varying lengths, with and without correcting lengths. The x-axes of the figure are cumulative: length 20 includes sentences of length 0–20, while length 10 includes 0–10. It is worth noting that BLEU is a word-level metric, but the systems were built using BPE; so the sequences actually generated are longer than the x-axes would suggest.

The baseline system on sentences with 10 words or less still has relatively high BLEU scores—even for a beam of 1000. Though there is a slight drop in BLEU (less than 2), it is not nearly as severe as when looking at the entire test set (more than 20). When correcting for length with normalization or word reward, the problem nearly disappears when considering the entire test set, with reward doing slightly better. For comparison, the rightmost points in each of the subplots correspond to the BLEU scores in columns 10 and 1000 of Table 1. This suggests that the beam problem is strongly related to the brevity problem.

#### 4.2.5 Length ratio

The interaction between the length problem and the beam problem can be visualized in the histograms of Figure 4 on the Russian–English system. In the upper left plot, the uncorrected model with beam 10 has the majority of the generated

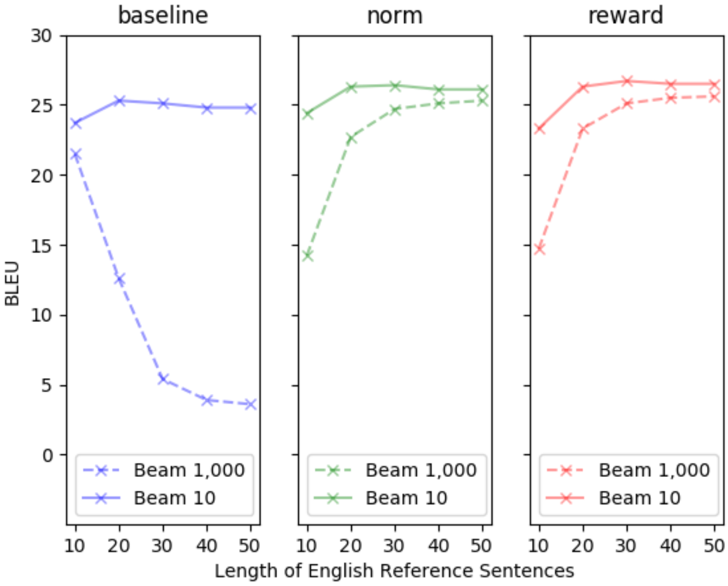


Figure 3: Impact of beam size on BLEU score when varying reference sentence lengths (in words) for Russian–English. The x-axis is cumulative moving right; length 20 includes sentences of length 0-20, while length 10 includes 0-10. As reference length increases, the BLEU scores of a baseline system with beam size of 10 remain nearly constant. However, a baseline system with beam 1000 has a high BLEU score for shorter sentences, but a very low score when the entire test set is used. Our tuned reward and normalized models do not suffer from this problem on the entire test set, but take a slight performance hit on the shortest sentences.

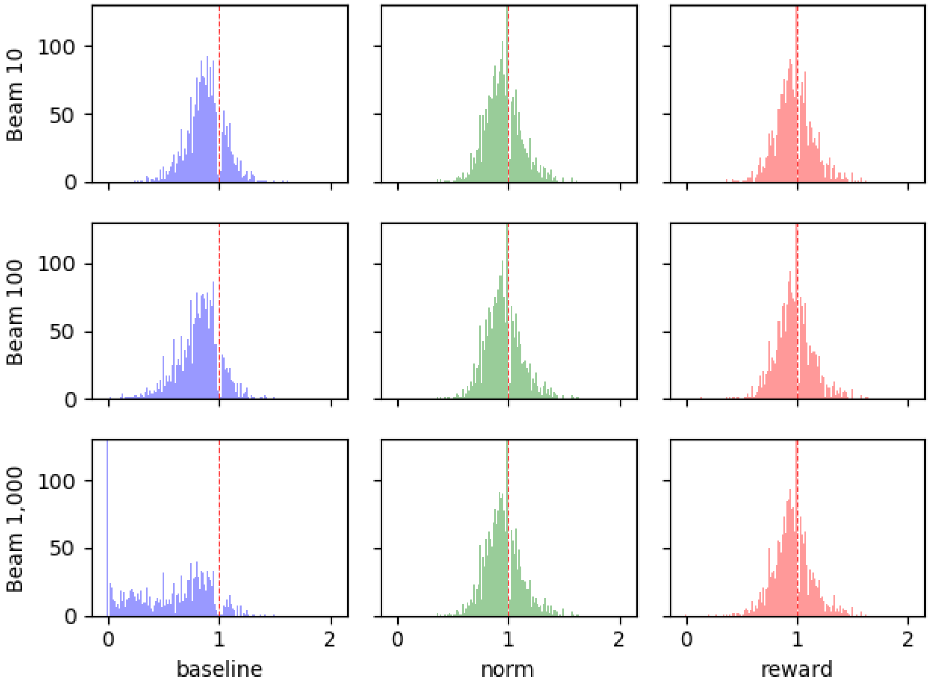


Figure 4: Histogram of length ratio between generated sentences and gold varied across methods and beam size for Russian–English. Note that the baseline method skews closer 0 as the beam size increases, while our other methods remain peaked around 1.0. There are a few outliers to the right that have been cut off, as well as the peaks at 0.0 and 1.0.

sentences with a length ratio close to 1.0, the gold lengths. Going down the column, as the beam size increases, the distribution of length ratios skews closer to 0. By a beam size of 1000, 37% of the sentences have a length of 0. However, both the word reward and the normalized models remain very peaked around a length ratio of 1.0 even as the beam size increases.

### 4.3 Tuning word reward

Above, we have shown that fixing the length problem with a word reward score fixes the beam problem. However these results are contingent upon choosing an adequate word reward score, which we have done in our experiments by optimization using a perceptron loss. Here, we show the sensitivity of systems to the value of this penalty, as well as the fact that there is not one correct penalty for all tasks. It is dependent on a myriad of factors including, beam size, dataset, and language pair.

#### 4.3.1 Sensitivity to $\gamma$

In order to investigate how sensitive a system is to the reward score, we varied values of $\gamma$ from 0 to 1.2 on both our German–English and Russian–English systems with a beam size of 50. BLEU scores and length ratios on 1000 heldout development sentences are shown in Figure 5. The length ratio is correlated with the word reward as expected, and the BLEU score varies by more than 5 points for German–English and over 4.5 points for Russian–English. On German–English, our method found a value of $\gamma = 0.57$, which is slightly higher than optimal; this is because the heldout sentences have a slightly shorter length ratio than the training sentences. Conversely, on Russian–English, our found value of $\gamma = 0.64$ is slightly lower than optimal as these heldout sentences have a slightly higher length ratio than the sentences used in training.

#### 4.3.2 Optimized $\gamma$ values

Tuning the reward penalty using the method described in Section 3.2 resulted in consistent improvements in METEOR scores and length ratios across all of our systems and language pairs. Tables 1, 2, and 3 show the optimized value of $\gamma$ for each beam size. Within a language pair, the optimal value of $\gamma$ is different for every beam size. Likewise, for a given beam size, the optimal value is different for every system. Our French–English and English–French systems in Table 3 have the exact same architecture, data, and training criteria. Yet, even for the same beam size, the tuned word reward scores are very different.

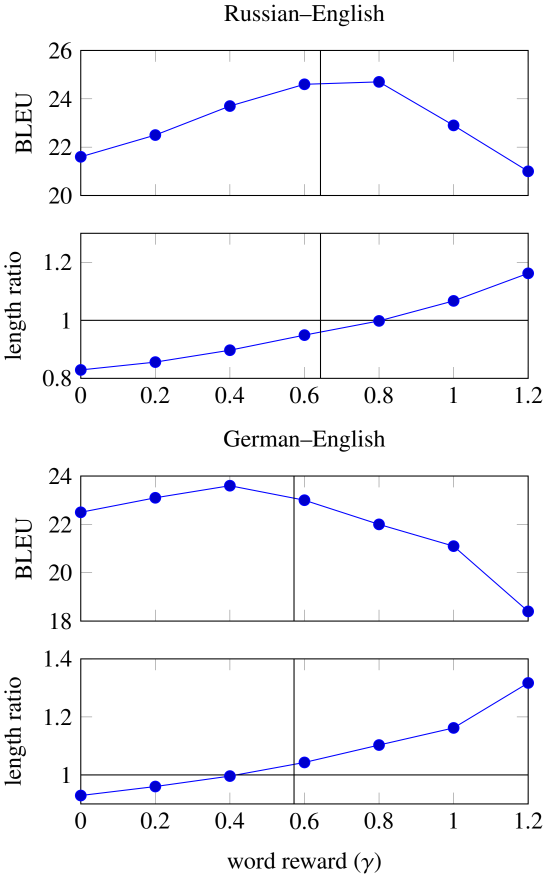


Figure 5: Effect of word penalty on BLEU and hypothesis length for Russian–English (top) and German-English (bottom) on 1000 unseen dev examples with beams of 50. Note that the vertical bars represent the word reward that was found during tuning.

**Training dataset size** Low-resource neural machine translation performs significantly worse than high-resource machine translation (Koehn and Knowles, 2017). Table 5 looks at the impact of training data size on BLEU scores and the beam problem by using 10% and 50% of the available Russian–English data. Once again, the optimal value of $\gamma$ is different across all systems and beam sizes. Interestingly, as the amount of training data decreases, the gains in BLEU using a tuned reward penalty increase with larger beam sizes. This suggests that the beam problem is more prevalent in lower-resource settings, likely due to the fact that less training data can increase the effects of label bias.

| Russian–English (medium)<br>Dataset Size | | Beam Size<br>10 | <br>50 | <br>75 | <br>100 | <br>150 |
|---|---|---|---|---|---|---|
| | baseline | 24.9 | 23.8 | 23.6 | 23.3 | 22.5 |
| 100% | reward | 26.5 | 26.6 | 26.5 | 26.5 | 26.5 |
| | $\gamma$ | 0.716 | 0.643 | 0.640 | 0.633 | 0.617 |
| | baseline | 22.8 | 21.4 | 20.8 | 20.4 | 19.2 |
| 50% | reward | 24.7 | 25.0 | 24.9 | 24.9 | 25.0 |
| | $\gamma$ | 0.697 | 0.645 | 0.638 | 0.636 | 0.646 |
| | baseline | 17.0 | 16.2 | 15.8 | 15.6 | 15.1 |
| 10% | reward | 17.6 | 18.0 | 18.0 | 18.0 | 18.1 |
| | $\gamma$ | 0.892 | 0.835 | 0.773 | 0.750 | 0.800 |

Table 5: Varying the size of the Russian–English training dataset results in different optimal word reward scores ($\gamma$). In all settings, the tuned score alleviates the beam problem. As the datasets get smaller, using a tuned larger beam improves the BLEU score over a smaller tuned beam. This suggests that lower-resource systems are more susceptible to the beam problem.

#### 4.3.3 Tuning time

Fortunately, the tuning process is very inexpensive. Although it requires decoding on a development dataset multiple times, we only need a small dataset. The time required for tuning our French–English and German–English systems is shown in Table 4. These experiments were run on an Nvidia GeForce GTX 1080Ti. The tuning usually takes a few minutes to hours, which is just a fraction of the overall training time. We note that there are numerous optimizations that could be taken to speed this up even more, such as storing the decoding lattice for partial reuse. However, we leave this for future work.

### 4.4 Word reward vs. length normalization

Tuning the word reward score generally had higher METEOR scores than length normalization across all of our settings. With BLEU, length normalization beat the word reward on German-English and French–English, but tied on English-French and lost on Russian–English. For the largest beam of 1000, the tuned word reward had a higher BLEU than length normalization. Overall, the two methods have relatively similar performance, but the tuned word reward has the more theoretically justified, globally-normalized derivation – especially in the context of label bias' influence on the brevity problem.

## 5 Conclusion

We have explored simple and effective ways to alleviate or eliminate the beam problem. We showed that the beam problem can largely be explained by the brevity problem, which results from the locally-normalized structure of the model. We compared two corrections to the model and introduced a method to learn the parameters of these corrections. Because this method is helpful and easy, we hope to see it included to make stronger baseline NMT systems.

We have argued that the brevity problem is an example of label bias, and that the solution is a very limited form of globally-normalized model. These can be seen as the simplest case of the more general problem of label bias and the more general solution of globally-normalized models for NMT (Wiseman and Rush, 2016; Venkatraman et al., 2015; Ranzato et al., 2015; Shen et al., 2016). Some questions for future research are:

- Solving the brevity problem leads to significant BLEU gains; how much, if any, improvement remains to be gained by solving label bias in general?
- Our solution to the brevity problem requires globally-normalized training on only a small dataset; can more general globally-normalized models be trained in a similarly inexpensive way?

## Acknowledgements

This research was supported in part by University of Southern California, subcontract 67108176 under DARPA contract HR0011-15-C-0115, and an Amazon Research Award to Chiang.

## References

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proc. ACL*, pages 2442–2452.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proc. Conference on Machine Translation*, pages 169–214.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 Conference on Machine Translation. In *Proc. Conference on Machine Translation*, volume 2, pages 131–198.

Léon Bottou. 1991. *Une Approche théorique de l'Apprentissage Connexioniste; Applications à la reconnaissance de la Parole*. Ph.D. thesis, Université de Paris Sud.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2013. Audio chord recognition with recurrent neural networks. In *ISMIR*, pages 335–340. Citeseer.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proc. Workshop on Statistical Machine Translation*.

Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with SMT features. In *Proc. AAAI*.

Liang Huang, Kai Zhao, and Mingbo Ma. 2017. When to finish? optimal beam search for neural text generation (modulo beam size). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2134–2139.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for wmt'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, pages 282–289.

Graham Neubig. 2015. lamtram: A toolkit for language and translation modeling using neural networks. http://www.github.com/neubig/lamtram.

Graham Neubig. 2016. Lexicons and minimum risk training for neural machine translation: NAIST-CMU at WAT2016. In *Proceedings of the 3rd Workshop on Asian Translation*, pages 119–125.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL*, pages 160–167.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. ACL*, pages 295–302. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318.

Michael Paul, Marcello Federico, and Sebastian Stüker. 2010. Overview of the IWSLT 2010 evaluation campaign. In *International Workshop on Spoken Language Translation (IWSLT) 2010*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. In *Proceedings of ICLR*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. ACL*, pages 1715–1725.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proc. of ACL*.

Arun Venkatraman, Martial Hebert, and J Andrew Bagnell. 2015. Improving multi-step prediction of learned time series models. In *AAAI*, pages 3024–3030.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of EMNLP*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Yue Zhang and Joakim Nivre. 2012. Analyzing the effect of global learning and beam-search on transition-based dependency parsing. In *Proceedings of COLING 2012: Posters*, pages 1391–1400.

# Extracting In-domain Training Corpora for Neural Machine Translation Using Data Selection Methods

**Catarina Cruz Silva**
Unbabel
R. Visc. de Santarém 67B,
1000-286 Lisboa, Portugal
catarina@unbabel.com

**Chao-Hong Liu, Alberto Poncelas, Andy Way**
ADAPT Centre, School of Computing,
Dublin City University
Dublin 9, Ireland
{chaohong.liu, alberto.poncelas, andy.way}@adaptcentre.ie

## Abstract

Data selection is a process used in selecting a subset of parallel data for the training of machine translation (MT) systems, so that 1) resources for training might be reduced, 2) trained models could perform better than those trained with the whole corpus, and/or 3) trained models are more tailored to specific domains. It has been shown that for statistical MT (SMT), the use of data selection helps improve the MT performance significantly. In this study, we reviewed three data selection approaches for MT, namely Term Frequency–Inverse Document Frequency, Cross-Entropy Difference and Feature Decay Algorithm, and conducted experiments on Neural Machine Translation (NMT) with the selected data using the three approaches. The results showed that for NMT systems, using data selection also improved the performance, though the gain is not as much as for SMT systems.

## 1 Introduction

Data selection is a technology used to improve Machine Translation (MT) performance by choosing a subset of the corpus for the training of MT systems (Chen et al., 2016). There are additional benefits using subsets instead of the whole corpus for MT training. Firstly, the training time could be reduced significantly. In some application scenarios, a much shorter training time would be very useful. Secondly, we could select data with the aim to make trained systems perform well for specific domains. In MT, models built with in-domain data perform better, as the vocabulary and sentence structures used in one domain (e.g. legal) differs from another unrelated domain (e.g. biotechnology).

There are several studies on data selection methods for SMT, showing good improvements over the baselines in which the whole corpora were used for training (Chen et al., 2016). A popular data selection method is cross-entropy difference (CED) (Moore and Lewis, 2010). In particular its bilingual variant (Axelrod et al., 2011) showed a positive impact of data selection for MT.

Term Frequency-Inverse Document Frequency (TF-IDF) (Salton and Yang, 1973) has also been used as a baseline data selection method in the literature. Data selection with cleaning was proposed to improve the robustness of training with divergent sentences (Carpuat et al., 2017).

Feature Decay Algorithms (FDA) are data selection methods that try to extract the subset of sentences by which the coverage of target language features is maximized (Biçici and Yuret, 2011). It has been used to select sentences from parallel data for SMT and NMT (Poncelas et al., 2018) in order to obtain a subset of data that is more tailored to a given test set.

Most of these results focused on comparing training of models from scratch for use in specific domains. The aforementioned papers do not include a focus on the impact of such techniques in finetuning the resulting trained model, which could be useful in the case where a baseline model works as an initialization and can be reused for any domain and thus reduce the time required to train the models for specific domains (van der Wees et al., 2017).

In this paper we evaluate the impact of data selection methods on Neural Machine Translation (NMT) systems. We would like to answer the following questions: Do data selection approaches improve domain NMT performance? Which of the three commonly used methods delivers the best results on data selection for NMT? How does the size of the seed and the selected training sentences affect the performance?

The paper is organised as follows. In Section 2, we give an overview of data selection approaches.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 224–231
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64023

Experimental setup and results are presented in Section 3 and Section 4. Conclusions and future work are given in Section 5.

## 2 Data Selection Methods

In order to train an MT model for a specific domain, it is best to use those sentences in a data set that are the most related to that domain. We use different data selection techniques to retrieve the sentences. These techniques aim to extract a subset of data from large datasets. The application of these techniques can be used to limit the amount of resource consumption, removing noise and/or adapting the data to a particular domain.

Among different data selection techniques (Eetemadi et al., 2015), in this work, we focus on three particular methods: Cross Entropy Difference (Section 2.1), TF-IDF Data Selection (Section 2.2), and Feature Decay Algorithms (Section 2.3).

### 2.1 Cross Entropy Difference

The Cross Entropy Difference method was first introduced by (Moore and Lewis, 2010) as a way to build more accurate in-domain Language Models for use in several tasks. The method is a variant of scoring by perplexity, since cross-entropy and perplexity are tightly coupled as shown in 1, where $b$ is the used base.

$$b^{-\sum_x \cdot p(x) \log q(x)} = b^{H(p,q)} \tag{1}$$

Given a general language model $LM_G$, built with out-of-domain data, and an in-domain language-model $LM_D$, the method ranks sentences $s$ using the cross-entropy difference in both language models, as in (2):

$$CED(s) = H_D(s) - H_G(s) \tag{2}$$

Although different ranking methods have been introduced, this method still remains popular among data selection approaches, having been used in recent work such as for the selection of monolingual data (Junczys-Dowmunt and Grundkiewicz, 2016), and for the selection of conversational data (Lewis and Federmann, 2015). Some work was also published on the use of neural language models for this purpose, such as Duh et al. (2013), but this applied to Statistical Machine Translation.

In our experiments, we built $n$-gram language models of order 5 using the KenLM tool[1] (Heafield, 2011). We then use the language model probability scores normalized by sentence length to compute the cross-entropy difference and rank the entire generic corpus.

### 2.2 TF-IDF data selection

The TF-IDF (Salton and Yang, 1973) method is widely known for its use in several information retrieval applications. It is defined in (3), where $\mathrm{tf}_{t,d}$ is the term frequency in the document, i.e. the ratio between the number of times the term appears in the sentence and the total number of terms, and $\mathrm{idf}_{t,d}$ is the inverse document frequency, the ratio between the total number of documents and the number of documents containing the term.

$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \cdot \frac{N}{\mathrm{df}_t} \tag{3}$$

To compute the TF-IDF measure in our experiments, we apply tokenization, remove punctuation and common stopwords in the texts, and finally truecase the sentences. We then consider every sentence in the domain corpus as a query sentence, and every sentence in the generic corpus as a document. Then, we obtain for each query a ranking of the documents, computed with cosine-similarity.

This ranking is stored for every query sentence and used to retrieve the K-nearest neighbours (KNN) necessary to obtain different data selection sizes.

### 2.3 Feature Decay Algorithms

Feature Decay Algorithms (FDA) (Biçici and Yuret, 2011; Biçici, 2013) are methods of data selection that try to extract, from a set of sentences, those that better represent a seed. It has been used in SMT to extract sentences from parallel corpora in order to obtain a subset of data more adapted to a given test set. These methods select sentences based on two criteria: a) the similarity with the seed (the more sequence of words it shares with the seed the better); and b) the variability of the words (the occurrences of the words shared with the seed should be well distributed, and avoid having too many occurrences of a few words).

These algorithms extract the $n$-grams from the seed as features. Each feature is assigned an initial value, indicating the relevance of being selected, and the sentences are scored as the normalized sum of values of contained features. Then, the sentences are iteratively selected. Each time a sentence is selected, the values of contained features

[1] https://github.com/kpu/kenlm

are decayed. Accordingly, it promotes selecting features that have not been previously selected in the process.

The decay function is defined in Equation (4):

$$decay(f) = init(f)\frac{d^{C_L(f)}}{(1 + C_L(f))^c} \quad (4)$$

where $L$ is the set of selected sentences and $C_L(f)$ is the count of the feature $f$ in $L$. $init(f)$ is an initialization function. The variables $d \in (0, 1]$ and $c \in [0, \infty)$ are parameters that regulate how much the value of the feature $f$ should decay. These values are by default (Biçici and Yuret, 2011) 0.5 and 0.0 for $d$ and $c$, respectively (so, by using default values the decay function in Equation (4) is $decay(f) = init(f)0.5^{C_L(f)}$). There are alternative ways of setting the values (Poncelas et al., 2016, 2017) that can obtain better results. However, in this work we used the default configuration of $d = 0.5$, $c = 0.0$ and used trigrams as features.

# 3 Experimental Setup

## 3.1 Data description

For the experiments we use English–French parallel data from two different domains/corpora: EMEA[2] and DGT[3] from the Open Parallel Corpus (OPUS) (Tiedemann, 2009). The first consists of medical data and the second a translation memory in the legal domain. We chose these domains in particular because they are categories more distant from the generic data, which is comprised of news data. The MultiUN corpus (Ziemski et al., 2016) is used for the training of generic models. Moreover, we use only its 6-way subset corpora, to be able to run the experiments in a more comparable setting.

## 3.2 Seed preparation

Although each data selection method has provided its own approach to select subsets from large corpora, in practice they would better perform if given a good initial subset (i.e. seed) to start with.

To prepare such an initial seed (the same seed is used in the three data selection algorithms), we remove noisy sentences considering punctuation and numerical character. In particular, we remove sentences where:

1. a source (or target) sentence contains fewer than $t_{chars}$ non-punctuation characters,
2. a source (or target) sentence contains fewer than $t_{words}$ words,
3. the source (or target) sentence ratio between punctuation characters and non-punctuation characters is above $t_{ratio}$.

where $t_{chars}$, $t_{words}$ and $t_{ratio}$ are thresholds. For both domains and language pairs, $t_{chars}$=5, $t_{words}$=2 and $t_{ratio}$=0.5 are used. We then removed duplicates using the source as reference and compile the remaining sentences into three parts: a validation set (2000 lines); a test set (2000 lines); and the remaining lines comprise the seed domain data. The EMEA domain corpus gave rise to a seed with 238K lines, and the DGT was truncated to a similar size, 250K, to keep experiments comparable.

## 3.3 Neural Machine Translation

The aim of this work is to assess the impact of data selection techniques on NMT. For this purpose, we use the Marian framework[4] (Junczys-Dowmunt et al., 2018) to train models using the attention-based encoder–decoder architecture as described in Sennrich et al. (2017).

For all experiments a preprocessing routine similar to the one in Moses[5] (Koehn et al., 2007) is used. The preprocessing consists of the following steps: entity replacement (on numbers, emails, urls and alphanumeric entities), tokenisation, truecasing and Byte-Pair Encoding (BPE) (Sennrich et al., 2016) with 89,500 merge operations.

# 4 Experiments

We present MT results using the three data selection methods and then use the best of the three methods to conduct a series of experiments to assess the impact of data selection on NMT models. We present two evaluation scores, BLEU (Papineni et al., 2002) and Translation Error Rate (TER) (Snover et al., 2006), in the tables. These scores give an estimation of how good the translation is: For BLEU, higher scores indicate better translations, while for TER, as it measures an error rate, lower scores indicate better translation performance.

We performed three different experiments:

- A comparison of the three data selection methods introduced in this paper (Section 4.1).

[2] http://opus.nlpl.eu/EMEA.php

[3] http://opus.nlpl.eu/DGT.php

[4] https://marian-nmt.github.io/

[5] http://www.statmt.org/moses/

| | TF-IDF | | CED | | FDA | |
|---|---|---|---|---|---|---|
| | BLEU ↑ | TER ↓ | BLEU ↑ | TER ↓ | BLEU ↑ | TER ↓ |
| Seed | .384 | .535 | .384 | .535 | .384 | .535 |
| + 240K (1:1) | .417 | .506 | .409 | .513 | **.439** | **.487** |
| + 480K (1:1) | .433 | .497 | .422 | .497 | **.441** | **.484** |
| + 480K (2:1) | .453 | .470 | .443 | .483 | **.464** | **.467** |
| + 1M (1:1) | .443 | .477 | .433 | .493 | **.445** | **.476** |
| + 1M (4:1) | .466 | .470 | .456 | .473 | **.477** | **.457** |
| + 2M (1:1) | .449 | .479 | .440 | .483 | **.452** | **.469** |
| + 2M (8:1) | .488 | **.445** | .479 | .453 | **.491** | .446 |

**Table 1:** Results of running three different data selection methods on different selection sizes for EMEA EN→FR. Both BLEU and TER are presented. The top result for each slice of selected data is presented in bold.

- A comparison of the data selection methods using different seeds (Section 4.2).
- The impact of the best data selection method in NMT (Section 4.3)

### 4.1 Comparison of methods

We start by comparing the three methods for the EMEA domain for English–French. Several experiments are run with different data selection sizes, between 250K and 2M lines, from the MultiUN corpus. We create different sizes of selected data in between these values, corresponding to a factor of 1, 2, 4 and 8 in relation to the size of the original seed. The comparison is not extended to larger selection sizes since a bigger slice, for example 4M, would already represent almost half of the total data available.

Table 1 shows the results of the three methods for models trained from scratch using seed data and different selected data. We present two approaches of combining the data. The first is a simple concatenation of the seed and the selected data. The second tries to balance the seed and the selected data in terms of the number of sentences used for training, by oversampling the seed a number of times such that there are approximately the same number of sentences in the selected data.

Two visible outcomes are shown in these experiments. The first is the overall gain of the Feature Decay Algorithm technique over its two counterparts. For every test (corresponding to a line in the table), the BLEU scores are better using the FDA method, followed by TF-IDF, with the CED method showing lower NMT performance. This result is interesting, since CED is one of the most common used methods for data selection and it has shown good results in several data selection experiments. However, these results are typically related to SMT, and in fact previous work in data selection has shown that these methods do not achieve the same performance for NMT.

The second result is that best performance was obtained when balancing the seed data with the selected data. We use this knowledge to guide the following experiments. Finally, in all experiments TER is also computed, and the results are consistent with those shown in BLEU scores.

### 4.2 Seed size variation

In previous experiments we used all the domain data available that passed our quality threshold, described in Section 3.2, and selected from the MultiUN corpus, which has little relation to the domain data. We conduct further experiments to analyse whether the previous results are dependent on the initial seed size and also to what extend the seed size impacts or limits the data selection gains.

We start with a seed of about 240K lines. To study the impact of the seed size we retrieve two subsets from the original seed with 50K lines and 100K lines. For each subset, we randomly sample the amount of lines from the original seed three different times and keep only the best subset, where the quality is evaluated by running a baseline MT experiment. Taking advantage of this preliminary experiment, we guarantee that the seed we choose from is not the worst to start with, increasing the reliability of these experiments.

Regarding our first goal, we can conclude that the previous results are not dependent on the initial seed size, from the results presented in Table 2, which consistently show that FDA performs best for all seeds. All experiments were run using balanced data since this showed enhanced perfor-

|  | TF-IDF |  | CED |  | FDA |  |
|---|---|---|---|---|---|---|
|  | BLEU ↑ | TER ↓ | BLEU ↑ | TER ↓ | BLEU ↑ | TER ↓ |
| 240K | .384 | .535 | .384 | .535 | .384 | .535 |
| + 240K | .417 | .506 | .409 | .513 | **.439** | **.487** |
| + 480K | .453 | .470 | .443 | .483 | **.464** | **.467** |
| + 1M | .466 | .470 | .456 | .473 | **.477** | **.457** |
| 100K | .306 | .613 | .306 | .613 | .306 | .613 |
| + 240K | .394 | **.520** | .361 | .550 | **.396** | .523 |
| + 480K | .419 | .507 | .396 | .533 | **.425** | **.498** |
| + 1M | .430 | .498 | .420 | .505 | **.432** | **.489** |
| 50K | .219 | .685 | .219 | .685 | .219 | .685 |
| + 240K | .368 | .554 | .296 | .633 | **.370** | **.552** |
| + 480K | .379 | .545 | .339 | .595 | **.390** | **.537** |
| + 1M | .391 | **.531** | .384 | .547 | **.394** | .535 |

**Table 2:** Results of running different data selection methods on different seed sizes for EMEA EN→FR. The top result for each seed size and slice of data selected is presented in bold. The ratio in the parentheses indicate the number of times seed was oversampled

mance, as mentioned in the previous section.

For the impact of the seed size on the data selection gains, the results show that for similar selected data, the score decreases with the seed, which is visible from the seed score to the 1M data selection. This is an intuitive result, since the amount of information contained in the full size seed is obviously larger than its counterparts.

However, it also shows that the gains from the baseline to the data selection are actually bigger for smaller seeds, with around 5–9 BLEU points increase for the full seed, 9–13 for the 100K sample and 16–18 points for the smaller 50K sample. This is consistent with the fact that the amount of data used has a bigger impact in NMT, especially when compared with previous knowledge about these methods in SMT.

### 4.3 Impact of data selection in NMT

Using the previous results as starting points, we focus now only on the FDA method for data selection and use oversampling of the seed to obtain a balanced training set.

#### 4.3.1 Full training

Several experiments are run for both domains, EMEA and DGT. To increase the confidence in our results, we repeat the experiment for English-Spanish, by selecting the corresponding Spanish sentences in both domain datasets.[6] All experiments for each language pair share the same seed data, oversampled to obtain a balanced corpus.

The results presented in Table 3 seem to support some of the previous conclusions that data selection does not yield as much gain for the NMT as it did for SMT. The best results are mostly data selection of 2M or 4M. However, the values are very close to the baseline obtained with the entire MultiUN data combined with the seed, which is balanced in the same way as the data selection methods. The results with 6M are also very close or slightly higher than the baseline, showing that more data helps almost as much as selected data.

#### 4.3.2 Adaptation from generic models

To try and separate the impact of the huge amount of data the generic model represents, we ran the same experiments in a fine-tuning scenario. In this context, a model is firstly trained with all the generic data until convergence, without any added domain knowledge. Then, a new training pass is ran until convergence with the domain data, where we add the selected data to the seed as pseudo-domain data. We mean to compare these selections with a baseline using only the seed, since using the full data here is redundant.

The data selection performed in the fine-tuning scenario has a negative impact, as shown in Table 4, where most of the data selection sets used obtain scores lower than the original seed baseline. One possible factor is that the MultiUN data contains very little domain data. As mentioned in the previous section, this experiment would gain from

[6] Both the DGT and EMEA datasets are available in EN–FR, EN–ES, and ES–FR, where part of the lines are aligned across the three languages.

| | EMEA$_{\text{EN}\rightarrow\text{FR}}$ | | DGT$_{\text{EN}\rightarrow\text{FR}}$ | | EMEA$_{\text{EN}\rightarrow\text{ES}}$ | | DGT$_{\text{EN}\rightarrow\text{ES}}$ | |
|---|---|---|---|---|---|---|---|---|
| | BLEU ↑ | TER ↓ | BLEU ↑ | TER ↓ | BLEU ↑ | TER ↓ | BLEU ↑ | TER ↓ |
| Seed | .384 | .535 | .427 | .469 | .432 | .485 | .413 | .453 |
| + 250K | .439 | .487 | .438 | .436 | .486 | .434 | .458 | .410 |
| + 500K | .464 | .467 | .464 | .417 | .511 | .418 | .476 | .397 |
| + 1M | .477 | .457 | .472 | .409 | .525 | .403 | .494 | .382 |
| + 2M | .491 | .446 | .482 | **.403** | .531 | **.396** | **.496** | .383 |
| + 4M | **.492** | **.441** | .478 | .404 | **.535** | .398 | .495 | **.379** |
| + 6M | .489 | .448 | .434 | .453 | .534 | .399 | .494 | .385 |
| + all data (11M) | .487 | .454 | **.482** | .405 | .495 | .449 | .493 | .384 |

**Table 3:** BLEU and TER scores for NMT training with different slices of selected data, using FDA for data selection. The top two results for each column are shaded, with the top result presented in bold

gathering a larger and more diverse generic corpus.

Moreover, all fine-tuning results are below the fully trained models with all data from the previous section. The most important factor here seems to be the highly technical vocabulary the models can have access to. While the model trained with all data has access to both the generic and domain vocabulary, the fine-tuned models are built on top of the generic vocabulary only. Thus, the model's input vocabulary of the first contains the most relevant domain words, while in the second these are split into subwords, as would happen to rare words.

### 4.3.3 Human evaluation

We also conducted a human evaluation using Unbabel's quality control system. For each language pair, translation direction and domain, 150 sentences were chosen randomly for evaluation. We then shuffled the content and provided it to evaluators ( professional linguists) for Fluency and Adequacy assessment. This assessment is done by rating each sentence from 1 to 5, and then computing the average for each model. The evaluators were not provided with the information as to which model was used to generate sentences. The definitions of Fluency and Adequacy, as used by the Unbabel Quality Team, are as follows.

Fluency addresses the linguistic well-formedness and naturalness of the text. Fluency errors include grammar, spelling or unintelligible text, sentence structure and word order issues, etc. In sum, these errors affect the reading and the comprehension of the text. The evaluation is done on the resulting translations without revealing their source sentences to the evaluators, to avoid biasing Fluency scores.

Adequacy addresses the relationship of the target text to the source text and can only be assessed by providing both translations and their source sentences to the editors. In other words, Adequacy addresses the extent to which a target text accurately renders the meaning of a source text. Adequacy errors include changes in intended meaning, addition and omission of content or any type of mistranslation, etc. In sum, Adequacy measures if the target text accurately reflect the meaning conveyed in the source text (Way, 2018).

The results of human evaluation on Fluency and Adequacy are presented in Table 5. The figures in the table correspond to the top scores in Tables 3 and 4. The results show that with fine-tuning of the training of models, Fluency is improved, especially for the EMEA models. Adequacy is also significantly improved in both EN-to-FR and EN-to-ES models. It shows very clear that data selection does improve the performance of all MT systems evaluated in this paper, in both Adequacy and Fluency.

It was also shown in Table 4 and Table 5 that for EN-to-FR, BLEU .452 of MT translated French sentences approximately corresponds to Fluency 4.25, and for EN-to-ES, BLEU .485 of MT translated Spanish sentences approximately corresponds to Fluency 4.50. In the future, we would like to make more comparisons between human evaluation metrics, e.g. Adequacy and Fluency as defined by Unbabel Quality Team, with commonly used MT performance metrics, e.g. BLEU and TER.

## 5 Conclusions

In this paper, we reviewed three commonly used data selection methods, i.e. TF-IDF, CED and FDA, for NMT. These methods improve the performance significantly for SMT. The results showed that FDA outperformed the other two methods. Although the gain in MT performance is not as much as

|  | EMEA$_{EN\rightarrow FR}$ |  | DGT$_{EN\rightarrow FR}$ |  | EMEA$_{EN\rightarrow ES}$ |  | DGT$_{EN\rightarrow ES}$ |  |
|---|---|---|---|---|---|---|---|---|
|  | BLEU ↑ | TER ↓ | BLEU ↑ | TER ↓ | BLEU ↑ | TER ↓ | BLEU ↑ | TER ↓ |
| MultiUN | .208 | .699 | .338 | .528 | .247 | .657 | .361 | .495 |
| Seed | .438 | .481 | **.476** | **.413** | **.486** | **.432** | **.487** | **.388** |
| + 250K | .429 | .485 | .462 | .418 | .469 | .442 | .473 | .399 |
| + 500K | **.439** | **.476** | .462 | .416 | .471 | .438 | .476 | .396 |
| + 1M | .436 | .478 | .465 | .414 | .478 | .440 | .477 | .397 |

**Table 4:** Fine-tuning approach for NMT training with data selection. The top two results for each column are shaded, with the top result presented in bold

| Models trained |  | EMEA$_{EN\rightarrow FR}$ |  | DGT$_{EN\rightarrow FR}$ |  | EMEA$_{EN\rightarrow ES}$ |  | DGT$_{EN\rightarrow ES}$ |  |
|---|---|---|---|---|---|---|---|---|---|
|  |  | AD ↑ | FL ↑ | AD ↑ | FL ↑ | AD ↑ | FL ↑ | AD ↑ | FL ↑ |
| From Scratch | Seed | 1.02 | 4.01 | 3.28 | 3.99 | 3.82 | 4.06 | 3.61 | 3.99 |
|  | + best slice | 4.18 | 3.95 | 3.87 | 4.39 | 4.25 | 4.42 | 4.22 | 4.50 |
|  | + all data (11M) | 4.1 | 3.95 | 3.78 | 4.29 | 3.99 | 4.33 | 4.19 | 4.47 |
| With Fine-tuning | Seed | 4.17 | 4.03 | 3.96 | 4.28 | 4.41 | 4.51 | 4.29 | 4.53 |
|  | + best slice | 4.22 | 4.05 | 4.12 | 4.45 | 4.43 | 4.50 | 4.30 | 4.52 |

**Table 5:** Human evaluation of Adequacy (AD) and Fluency (FL) for top scores in previous experiments in Tables 3 and 4

that in SMT systems, our experiments showed that using EMEA and MultiUN corpora, NMT systems trained with FDA-selected data still outperform systems trained with the whole corpus, in terms of both BLEU and TER.

In addition to using data selection, training with fine-tuning from pre-trained models is also employed to further improve MT performance. We conducted human evaluation by professional linguists, in which Adequacy and Fluency are assessed. The results show that models trained with selected data constantly outperformed those trained with the whole corpus, in both human evaluation measures. By employing fine-tuning on top of data selection, MT performance is further improved significantly in both Adequacy and Fluency.

## Acknowledgements

The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund. This project has partially received funding from the European Union's Horizon 2020 Research and Innovation programme under the Marie Skłodowska-Curie Actions (Grant No. 734211). We would also like to thank Dr André Martins, Dr Julie Belião and Ms Marianna Buchicchio from Unbabel AI and Quality Teams for their help with the human evaluation.

## References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Edinburgh, United Kingdom.

Ergun Biçici. 2013. Feature decay algorithms for fast deployment of accurate statistical machine translation systems. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 78–84, Sofia, Bulgaria.

Ergun Biçici and Deniz Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283, Edinburgh, Scotland.

Marine Carpuat, Yogarshi Vyas, and Xing Niu. 2017. Detecting cross-lingual semantic divergence for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 69–79, Vancouver, Canada.

Boxing Chen, Roland Kuhn, George Foster, Colin Cherry, and Fei Huang. 2016. Bilingual methods for adaptive training data selection for machine translation. In *The Twelfth Conference of The Association for Machine Translation in the Americas*, pages 93–106, Austin, Texas.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *The 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 678–683, Sofia, Bulgaria.

Sauleh Eetemadi, William Lewis, Kristina Toutanova, and Hayder Radha. 2015. Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29(3-4):189–223.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 751–758, Berlin, Germany.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for SMT. In *Proceedings of 45th annual meeting of the ACL on interactive poster & demonstration sessions*, pages 177–180, Prague, Czech Republic.

Will Lewis and Christian Federmann. 2015. Applying cross-entropy difference for selecting parallel training data from publicly available sources for conversational machine translation. In *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, pages 126–134, Da Nang, Vietnam.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. 2017. Applying n-gram alignment entropy to improve feature decay algorithms. *The Prague Bulletin of Mathematical Linguistics*, 108(1):245–256.

Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. 2018. Feature decay algorithms for neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 239–248, Alacant, Spain.

Alberto Poncelas, Andy Way, and Antonio Toral. 2016. Extending feature decay algorithms using alignment entropy. In *International Workshop on Future and Emerging Trends in Language Technology*, pages 170–182, Seville, Spain.

Gerard Salton and Chung-Shu Yang. 1973. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725, Berlin, Germany.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. 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*, pages 223–231, Cambridge, Massachusetts, USA.

Jörg Tiedemann. 2009. News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume 5, pages 237–248, Borovets, Bulgaria.

Andy Way. 2018. Quality expectations of machine translation. In Joss Moorkens, Sheila Castilho, Federico Gaspari, and Stephen Doherty, editors, *Translation Quality Assessment: From Principles to Practice*, volume 1, pages 159–178. Springer.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0. In *International Conference on Language Resources and Evaluation LREC*, pages 3530–3534, Portorož, Slovenia.

# Massively Parallel Cross-Lingual Learning in Low-Resource Target Language Translation

**Zhong Zhou**
Carnegie Mellon University
zhongzhou@cmu.edu

**Matthias Sperber**
Karlsruhe Institute of Technology
matthias.sperber@kit.edu

**Alex Waibel**
Carnegie Mellon University
Karlsruhe Institute of Technology
alex@waibel.com

## Abstract

We work on translation from rich-resource languages to low-resource languages. The main challenges we identify are the lack of low-resource language data, effective methods for cross-lingual transfer, and the variable-binding problem that is common in neural systems. We build a translation system that addresses these challenges using eight European language families as our test ground. Firstly, we add the source and the target family labels and study intra-family and inter-family influences for effective cross-lingual transfer. We achieve an improvement of +9.9 in BLEU score for English-Swedish translation using eight families compared to the single-family multi-source multi-target baseline. Moreover, we find that training on two neighboring families closest to the low-resource language is often enough. Secondly, we construct an ablation study and find that reasonably good results can be achieved even with considerably less target data. Thirdly, we address the variable-binding problem by building an order-preserving named entity translation model. We obtain 60.6% accuracy in qualitative evaluation where our translations are akin to human translations in a preliminary study.

## 1 Introduction

We work on translation from a rich-resource language to a low-resource language. There is usually little low-resource language data, much less parallel data available (Duong et al., 2016; Anastasopoulos et al., 2017); Despite of the challenges of little data and few human experts, it has many useful applications. Applications include translating water, sanitation and hygiene (WASH) guidelines to protect Indian tribal children against waterborne diseases, introducing earthquake preparedness techniques to Indonesian tribal groups living near volcanoes and delivering information to the disabled or the elderly in low-resource language communities (Reddy et al., 2017; Barrett, 2005; Anastasiou and Schäler, 2010; Perry and Bird, 2017). These are useful examples of translating a closed text known in advance to the low-resource language.

There are three main challenges. Firstly, most of previous works research on individual languages instead of collective families. Cross-lingual impacts and similarities are very helpful when there is little data in low-resource language (Shoemark et al., 2016; Sapir, 1921; Odlin, 1989; Cenoz, 2001; Toral and Way, 2018; De Raad et al., 1997; Hermans, 2003; Specia et al., 2016). Secondly, most of the multilingual Neural Machine Translation (NMT) works assume the same amount of training data for all languages. In the low-resource case, it is important to exploit low or partial data in low-resource language to produce high quality translation. The third issue is the variable-binding problem that is common in neural systems, where "John calls Mary" is treated the same way as "Mary calls John" (Fodor and Pylyshyn, 1988; Graves et al., 2014). It is more challenging when both "Mary" and "John" are rare words. Solving the binding problem is crucial because the mistakes in named entities change the meaning of the translation. It is especially challenging in the low-resource case because many words are rare words.

Our contribution in addressing these issues is three-fold, extending from multi-source multi-target attentional NMT. Firstly, to examine intra-family and inter-family influences, we add source and target language family labels in training. Training on multiple families improves BLEU score significantly; moreover, we find training on two neighboring families closest to the low-resource language gives reasonably good BLEU scores, and we define neighboring families closely in Section 3.2. Secondly, we conduct an ablation

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 232–243
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64024

study to explore how generalization changes with different amounts of data and find that we only need a small amount of low-resource language data to produce reasonably good BLEU scores. We use full data except for the ablation study. Finally, to address the variable-binding problem, we build a parallel lexicon table across twenty-three European languages and devise a novel method of order-preserving named entity translation method. Our method works in translation of any text with a fixed set of named entities known in advance. Our goal is to minimize manual labor, but not to fully automate to ensure the correct translation of named entities and their ordering.

In this paper, we begin with introduction and related work in Section 1 and 2. We introduce our methods in addressing three issues that are important for translation into low-resource language in Section 3.2, as proposed extensions to our baseline in Section 3.1. Finally, we present our results in Section 4 and conclude in Section 5.

## 2 Related Work

### 2.1 Multilingual Attentional NMT

Attentional NMT is trained directly in an end-to-end system and has flourished recently (Wu et al., 2016; Sennrich et al., 2016; Ling et al., 2015). Machine polyglotism, training machines to be proficient in many languages, is a new paradigm of multilingual NMT (Johnson et al., 2017; Ha et al., 2016; Firat et al., 2016; Zoph and Knight, 2016; Dong et al., 2015; Gillick et al., 2016; Al-Rfou et al., 2013; Tsvetkov et al., 2016). Many multilingual NMT systems involve multiple encoders and decoders, and it is hard to combine attention for quadratic language pairs bypassing quadratic attention mechanisms (Firat et al., 2016). In multi-source scenarios, multiple encoders share a combined attention mechanism (Zoph and Knight, 2016). In multi-target scenarios, every decoder handles its own attention with parameter sharing (Dong et al., 2015). Attention combination schemes include simple combination and hierarchical combination (Libovickỳ and Helcl, 2017).

The state-of-the-art of multilingual NMT is adding source and target language labels in training a universal model with a single attention scheme, and Byte-Pair Encoding (BPE) is used at preprocessing stage (Ha et al., 2016). This method is elegant in its simplicity and its advancement in low-resource language translation as well as zero-shot translation using pivot-based translation scheme (Johnson et al., 2017). However, these works have training data that increases quadratically with the number of languages (Dong et al., 2015; Gillick et al., 2016), rendering training on massively parallel corpora difficult.

### 2.2 Sub-word Level NMT

Many NMT systems lack robustness with out-of-vocabulary words (*OOV*s) (Wu et al., 2016). Most *OOV*s are treated as unknowns (*$UNK*s) uniformly, even though they are semantically important and different (Ling et al., 2015; Sennrich et al., 2016). To tackle the *OOV* problem, researchers work on byte-level (Gillick et al., 2016) and character-level models (Ling et al., 2015; Chung et al., 2016). Many character-level models do not work as well as word-level models, and do not produce optimal alignments (Tiedemann, 2012). As a result, many researchers shift to sub-word level modeling between character-level and word-level. One prominent direction is BPE which iteratively learns subword units and balances sequence length and expressiveness with robustness (Sennrich et al., 2016).

### 2.3 Lexiconized NMT

Much research is done in translating lexicons and named entities in NMT (Nguyen and Chiang, 2017; Wang et al., 2017; Arthur et al., 2016). Some researchers create a separate character-level named entity model and mark all named entities as *$TERM*s to train (Wang et al., 2017). This method learns people's names well but does not improve BLEU scores (Wang et al., 2017). It is time-consuming and adds to the system complexity. Other researchers attempt to build lexicon translation seamlessly with attentional NMT by using an affine transformation of attentional weights (Nguyen and Chiang, 2017; Arthur et al., 2016). Some also attempt to embed cross-lingual lexicons into the same vector space for transfer of information (Duong et al., 2017).

## 3 Translation System

### 3.1 Baseline Translation System

Our baseline is multi-source multi-target attentional NMT within one language family through adding source and target language labels with a single unified attentional scheme, with BPE used at the preprocessing stage. The source and target vocabulary are not shared.

| Families | Languages |
|---|---|
| Germanic | German (de) Danish (dn) Dutch (dt) Norwegian (no) Swedish (sw) English (en) |
| Slavic | Croatian (cr) Czech (cz) Polish (po) Russian (ru) Ukrainian (ur) Bulgarian (bg) |
| Romance | Spanish (es) French (fr) Italian (it) Portuguese (po) Romanian (ro) |
| Albanian | Albanian (ab) |
| Hellenic | Greek (gk) |
| Italic | Latin (ln) [descendants: Romance languages] |
| Uralic | Finnish (fn) Hungarian (hg) |
| Celtic | Welsh (ws) |

Table 1: Language families. Language codes are in brackets.

## 3.2 Proposed Extensions

We present our methods in solving three issues relevant to translation into low-resource language as our proposed extensions.

### 3.2.1 Language Families and Cross-lingual Learning

Cross-lingual and cross-cultural influences and similarities are important in linguistics (Shoemark et al., 2016; Levin et al., 1998; Sapir, 1921; Odlin, 1989; Cenoz, 2001; Toral and Way, 2018; De Raad et al., 1997; Hermans, 2003; Specia et al., 2016). The English word, “Beleaguer” originates from the Dutch word “belegeren”; “fidget” originates from the Nordic word “fikja”. English and Dutch belong to the same family and their proximity has effect on each other (Harding and Sokal, 1988; Ross et al., 2006). Furthermore, languages that do not belong to the same family affect each other too (Sapir, 1921; Ammon, 2001; Toral and Way, 2018). “Somatic” stems from the Greek word “soma”; “広告” (Japanese), “광고”(Korean), “Quảng cáo”(Vietnamese) are closely related to the Traditional Chinese word “廣告”. Indeed, many cross-lingual similarities are present.

In this paper, we use the language phylogenetic tree as the measure of closeness of languages and language families (Petroni and Serva, 2008). The distance measure of language families is the collective of all of the component languages. Language families that are next to each other in the language phylogenetic tree are treated as neighboring families in our paper, like Germanic family and Romance family. In our discussion in this paper, we will often refer to closely related families in the language phylogenetic tree as neighboring families.

We prepend the source and target family labels, in addition to the source and target language labels to the source sentence to improve convergence rate and increase translation performance. For example, all French-to-English translation pairs are prepended with four labels, the source and target family labels and the source and target languages labels, i.e., `__opt_family_src_romance` `__opt_family_tgt_germanic` `__opt_src_fr` `__opt_tgt_en`. In Section 4, we examine intra-family and inter-family effects more closely.

### 3.2.2 Ablation Study on Target Training data

To achieve high information transfer from rich-resource language to low-resource target language, we would like to find out how much target training data is needed to produce reasonably good performance. We vary the amount of low-resource training data to examine how to achieve reasonably good BLEU score using limited low-resource data. In the era of Statistical Machine Translation (SMT), researchers have worked on data sampling and sorting measures (Eck et al., 2005; Axelrod et al., 2011).

To rigorously determine how much low-resource target language is needed for reasonably good results, we do a range of control experiments by drawing samples from the low-resource language data randomly with replacement and duplicate them if necessary to ensure all experiments carry the same number of training sentences. We keep the amount of training data in rich-resource languages the same, and vary the amount of training data in low-resource language to conduct rigorous control experiments. Our data selection process is different from prior research in that only the low-resource training data is reduced, simulating the real world scenario of having little data in low-resource language. By comparing results from control experiments, we determine how much low-resource data is needed.

### 3.2.3 Order-preserving Lexiconized NMT

The variable-binding problem is an inherent issue in connectionist architectures (Fodor and Pylyshyn, 1988; Graves et al., 2014). “John calls Mary” is not equivalent to “Mary calls John”, but neural networks cannot distinguish the two easily (Fodor and Pylyshyn, 1988; Graves et al., 2014). The failure of traditional NMT to distinguish the subject and the object of a sentence is detrimental. For example, in the narration “John told his son Ryan to help David, the brother of Mary”, it is a serious mistake if we reverse John and Ryan’s father-son relationships or confuse Ryan’s and David’s

| lan | de | dn | dt | en | no | sw |
|---|---|---|---|---|---|---|
| de | N.A. | 37.5 | 43.4 | 45.1 | 41.1 | 35.8 |
| dn | 39.0 | N.A. | 37.1 | 41.1 | 42.6 | 37.4 |
| dt | 43.5 | 36.3 | N.A. | 45.1 | 39.0 | 34.3 |
| en | 40.4 | 34.5 | 41.1 | N.A. | 37.1 | 34.0 |
| no | 40.5 | 42.7 | 40.4 | 42.8 | N.A. | 40.6 |
| sw | 39.4 | 38.9 | 37.5 | 40.4 | 43.0 | N.A. |

Table 2: (Baseline model) Germanic family multi-source multi-target translation. Each row represents source, each column represents target. Language codes follow Table 1.

relationships with Mary.

In our research on translation, we focus mainly on text with a fixed set of named entities known in advance. We assume that experts help to translate a given list of named entities into low-resource language first before attempting to translate any text. Under this assumption, we propose an order-preserving named entity translation mechanism. Our solution is to first create a parallel lexicon table for all twenty-three European languages using a seed English lexicon table and fast-aligning it with the rest (Dyer et al., 2013). Instead of using *$UNK*s to replace the named entities, we use *$NE*s to distinguish them from the other unknowns. We also sequentially tag named entities in a sentence as *$NE1*, *$NE2*, . . . , to preserve their ordering. For every sentence pair in the multilingual training, we build a target named entity decoding dictionary by using all target lexicons from our lexicon table that matches with those appeared in the source sentence. During the evaluation stage, we replace all the numbered *$NE*s using the target named entity decoding dictionary to present our final translation. This method improves translation accuracy greatly and preserves the order.

As a result of our contribution, the experts only need to translate a few lexicons and a small amount of low-resource text before passing the task to our system to obtain good results. Post-editing and minor changes may be required to achieve 100% accuracy before the releasing the translation to the low-resource language communities.

## 4 Experiments and Results

We choose the Bible corpus as a test ground for our proposed extensions because the Bible is the most translated text that exists and is freely accessible. Though it has limitations, it does not have copyright issues like most of literary works that are translated into many languages do. There are many research works done using the Bible (Naaijer and Roorda, 1993; Mayer and Cysouw, 2014; Scannell, 2006; Dufter and Schütze, 2018; Resnik et al., 1999; Chan and Pollard, 2001; Banchs and Costa-Jussà, 2011; Christodouloupoulos and Steedman, 2015; Beale et al., 2005). Unlike many past research works where only New Testament is used (Dufter and Schütze, 2018), we use both Old Testament and New Testament in our Bible corpus. We align all Bible verses across different languages.

We train our proposed model on twenty-three European languages across eight families on a parallel Bible corpus. For our purpose, we treat Swedish as our hypothetical low-resource target language, English as our rich-resource language in the single-source single-target case and all other Germanic languages as our rich-resource languages in the multi-source multi-target case.

Firstly, we present our data and training parameters. Secondly, we add family tags in different configurations of language families showing intra-family and inter-family effects. Thirdly, we conduct an ablation study and plot the generalization curves by varying the amount of training data in Swedish, and we show that training on one fifth of the data give reasonably good BLEU scores. Lastly, we devise an order-preserving lexicon translation method by building a parallel lexicon table across twenty-three European languages and tagging named entities in order.

### 4.1 Data and Training Parameters

We clean and align the Bible in twenty-three European languages in Table 1. We randomly sample the training, validation and test sets according to the 0.75, 0.15, 0.10 ratio. Our training set contains 23K verses, but is massively parallel. In our control experiments, we also use the experiment training on the WMT'14 French-English dataset together with French and English Bibles to compare with our results. Note that our WMT baseline contains French and English Bibles in addition to the WMT'14 data, and is used to contrast our results with the effect of increasing data.

In all our experiments, we use a minibatch size of 64, dropout rate of 0.3, 4 RNN layers of size 1000, a word vector size of 600, learning rate of 0.8 across all LSTM-based multilingual experiments. For single-source single-target translation, we use 2 RNN layers of size 500, a word vector size of 500, and learning rate of 1.0. All learning rates are decaying at the rate of 0.7 if the validation score is not improving or it is past epoch 9. We use SGD as our learning algorithm. We build our

| expt | S | G | GS | GR | 3F | 8F |
|---|---|---|---|---|---|---|
| de2sw | 4.0 | 35.8 | 42.0 | 42.2 | 42.5 | 42.8 |
| dn2sw | 16.9 | 37.4 | 43.4 | 41.8 | 42.7 | 41.7 |
| dt2sw | 4.8 | 34.3 | 41.4 | 41.6 | 42.8 | 42.5 |
| en2sw | 6.9 | 34.0 | 40.3 | 40.2 | 41.8 | 42.1 |
| no2sw | 16.8 | 40.6 | 43.6 | 44.0 | 44.5 | 43.1 |

Table 3: Inter-family and intra-family effects on BLEU scores with respect to increasing addition of language families.
S: single-source single-target NMT.
G: training on Germanic family.
GS: training on Germanic, Slavic family.
GR: training on Germanic, Romance family.
3F: training on Germanic, Slavic, Romance family.
8F: training on all 8 European families together.

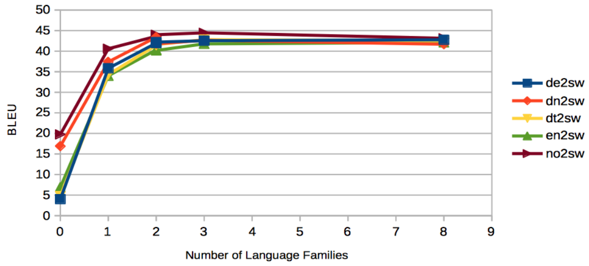


Figure 1: Intra-family and inter-family effects on BLEU scores with respect to increasing addition of language families.

code based on OpenNMT (Klein et al., 2017). For the ablation study, we train on BLEU scores directly until the *Generalization Loss* (*GL*) exceeds a threshold of $\alpha = 0.1$ (Prechelt, 1998). *GL* at epoch $t$ is defined as $GL(t) = 100(1 - \frac{E_{val}^{t}}{E_{opt}^{t}})$, modified by us to suit our objective using BLEU scores (Prechelt, 1998). $E_{val}^{t}$ is the validation score at epoch $t$ and $E_{opt}^{t}$ is the optimal score up to epoch $t$. We evaluate our models using both BLEU scores (Papineni et al., 2002) and qualitative evaluation.

### 4.2 Family labels and Intra-family & Inter-family Effects

We first investigate intra-family and inter-family influences and the effects of adding family labels. We use full training data in this subsection. Adding family labels not only improves convergence rate, but also increases BLEU scores.

**Languages have varying closeness to each other:** Single-source single-target translations of different languages in Germanic family to Swedish show huge differences in BLEU scores as shown in Table 3. These differences are well aligned with the multi-source multi-target results. Norwegian-Swedish and Danish-Swedish translations have much higher BLEU scores than the rest. This hints that Norwegian and Danish are closer to Swedish than the rest in the neural representation.

| expt | S | G | GSl | GRl | 3Fl | 8Fl |
|---|---|---|---|---|---|---|
| de2sw | 4.0 | 35.8 | 41.8 | 42.2 | 42.5 | 44.3 |
| dn2sw | 16.9 | 37.4 | 43.0 | 41.5 | 42.5 | 42.8 |
| dt2sw | 4.8 | 34.3 | 41.4 | 41.8 | 42.7 | 42.3 |
| en2sw | 6.9 | 34.0 | 40.9 | 40.4 | 41.7 | 43.9 |
| no2sw | 16.8 | 40.6 | 43.7 | 44.3 | 44.2 | 44.7 |

Table 4: Effects of adding family labels on BLEU scores with respect to increasing addition of language families.
S and G: same as in Table 3.
GSl: Germanic, Slavic family with family labels.
GRl: Germanic, Romance family with family labels.
3Fl: Germanic, Slavic, Romance family with family labels.
8Fl: all 8 European families together with family labels

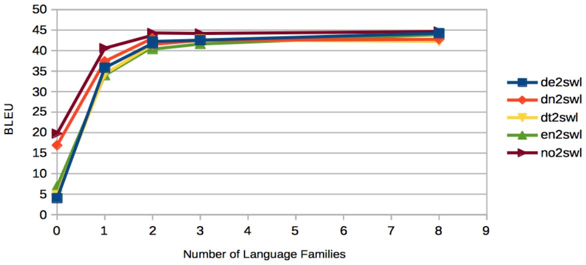


Figure 2: Effects of adding family labels on BLEU scores with respect to increasing addition of language families.

**Multi-source multi-target translation improves greatly from single-source single-target translation:** English-Swedish single-source single-target translation gives a low BLEU score of 6.9 as shown in Table 3, which is understandable as our dataset is very small. BLEU score for English-Swedish translation improves greatly to 34.0 in multi-source multi-target NMT training on Germanic family as shown in Table 2. In this paper, we treat Germanic multi-source multi-target NMT as our baseline model. Complete tables of multi-source and multi-target experiments are in the appendices. We present only relevant columns important for cross-lingual learning and translation into low-resource language here.

**Adding languages from other families into training improves translation quality within each family greatly:** English-Swedish translation's BLEU score improves significantly from 34.0 to 40.3 training on Germanic and Slavic families, and 40.2 training on Germanic and Romance families as shown in Table 3. After we add all three families in training, BLEU score for English-Swedish translation increases further to 41.8 in Table 3. Finally, after we add all eight families, BLEU score for English-Swedish translation increases to 42.1 in Table 3.

**A Plateau is observed after adding more than one neighboring family:** A plateau is observed when we plot Table 3 in Figure 1. The increase in BLEU scores after adding two families is much milder than that of the first addition of a neighbor-

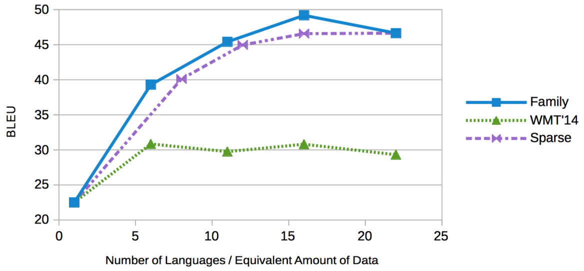


Figure 3: Comparison of different ways of increasing training Data in French-English translation.
Family: Adding data from other languages based on the family unit
WMT'14: Adding WMT'14 data as control experiment
Sparse: Adding data from other languages that spans the eight European families

ing family. This hints that using unlimited number of languages to train may not be necessary.

**Adding family labels not only improves convergence rate, but also increases BLEU scores:** We observe in Table 4 that BLEU scores for most language pairs improve with the addition of family labels. Training on eight language families, we achieve a BLEU score of 43.9 for English-Swedish translation, +9.9 above the Germanic baseline. Indeed, the more families we have, the more helpful it is to distinguish them.

**Training on two neighboring families nearest to the low-resource language gives better result than training on languages that are further apart:** Our observation of the plateau hints that training on two neighboring families nearest to the low-resource language is good enough as shown in Table 3. Before jumping to conclusion, we compare results of adding languages by family with that of adding languages by random samples that span all eight families, defined as the following.

**Definition 4.1** (Language Spanning)**.** A set of languages spans a set of families when it contains at least one language from each family.

In Figure 3, we conduct a few experiments on French-English translation using different ways of adding training data. Let *family addition* describe the addition of training data through adding close-by language families based on the unit of family; let *sparse addition* describe the addition of training data through adding language sets that spans eight language families. In sparse addition, languages are further apart as each may represent a different family. We find that family addition gives better generalization than that of sparse addition. It strengthens our earlier results that training on two families closest to our low-resource language is a reliable way to reach good generalization.

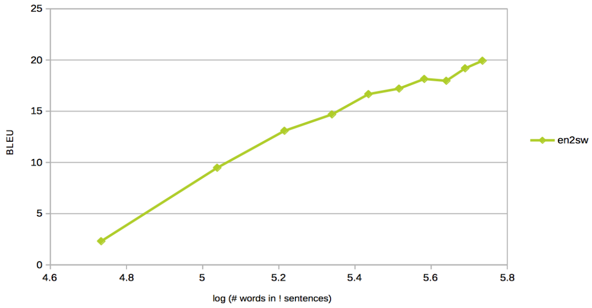


Figure 4: Single-source single-target English-Swedish BLEU plots against increasing amount of Swedish data.

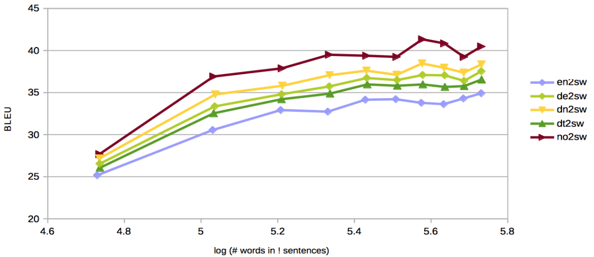


Figure 5: Multi-source multi-target Germanic-family-trained BLEU plots against increasing amount of Swedish data.

**Generalization is not merely an effect of increasing amount of data:** In Figure 3, we compare all methods of adding languages against a WMT'14 curve by using equivalent amount of WMT'14 French-English data in each experiment. The WMT'14 curve serve as our benchmark of observing the effect of increasing data, we observe that our addition of other languages improve BLEU score much sharply than the increase in the benchmark, showing that our generalization is not merely an effect of increasing data. We also observe that though increase WMT'14 data initially increases BLEU score, it reaches a plateau and adding more WMT'14 data does not increase performance from very early point.

## 4.3 Ablation Study on Target Training Data

We use full training data from all rich-resource languages, and we vary the amount of training data in Swedish, our low-resource language, spanning from one tenth to full length uniformly. We duplicate the subset to ensure all training sets, though having a different number of unique sentences, have the same number of total sentences.

**Power-law relationship is observed between the performance and the amount of training data in low-resource language:** Figure 5 shows how BLEU scores vary logarithmically with the number of unique sentences in the low-resource training data. It follows a linear pattern for single-source single-target translation from English to

| Data | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| #w | 53589 | 107262 | 161332 | 214185 | 268228 | 322116 | 375439 | 429470 | 483440 | 538030 |
| log#w | 4.73 | 5.03 | 5.21 | 5.33 | 5.43 | 5.51 | 5.57 | 5.63 | 5.68 | 5.73 |
| en2sw | 25.2 | 30.6 | 32.9 | 32.7 | 34.2 | 34.2 | 33.8 | 33.6 | 34.3 | 34.9 |
| de2sw | 26.5 | 33.4 | 34.8 | 35.7 | 36.7 | 36.5 | 37.1 | 37.1 | 36.4 | 37.5 |
| dn2sw | 27.2 | 34.8 | 35.8 | 37.1 | 37.6 | 37.1 | 38.5 | 38.0 | 37.4 | 38.4 |
| dt2sw | 26.1 | 32.5 | 34.2 | 34.9 | 36.0 | 35.8 | 36.0 | 35.7 | 35.8 | 36.6 |
| no2sw | 27.7 | 36.9 | 37.9 | 39.5 | 39.4 | 39.2 | 41.3 | 40.8 | 39.2 | 40.5 |

Table 5: Ablation Study on Germanic Family. #w is the word count of unique sentences in Swedish data.

| en | de | cz | es | fn | sw |
|---|---|---|---|---|---|
| Joseph | Joseph | Jozef | José | Joseph | Josef |
| Peter | Petrus | Petr | Pedro | Pietari | Petrus |
| Zion | Zion | Sion | Sion | Zionin | Sion |
| John | Johannes | Jan | Juan | Johannes | Johannes |
| Egypt | Ägypten | Egyptské | Egipto | Egyptin | Egyptens |
| Noah | Noah | Noé | Noé | Noa | Noa |

Table 6: A few examples from the parallel lexicon table.

| expt | G | OG | OG1 | OGM |
|---|---|---|---|---|
| de2sw | 35.8 | 36.6 | 36.6 | 36.9 |
| dn2sw | 37.4 | 37.0 | 37.2 | 36.9 |
| dt2sw | 34.3 | 35.8 | 35.6 | 35.9 |
| en2sw | 34.0 | 33.6 | 33.9 | 33.4 |
| no2sw | 40.6 | 41.2 | 41.0 | 41.4 |

Table 7: Summary of order-preserving lexicon translation.
G: training on Germanic family without using order-preserving method.
OG: order-preserving lexicon translation.
OG1: OG translation using lexicons with frequency 1.
OGM: OG translation using lexicons with manual selection.

Swedish as shown in Figure 4. We also observe a linear pattern for the multi-source multi-target case, though more uneven in Figure 5. The linear pattern with BLEU scores against the logarithmic data shows the power-law relationship between the performance in translation and the amount of low-resource training data. Similar power-law relationships are also found in past research and contemporary literature (Turchi et al., 2008; Hestness et al., 2017).

**We achieve reasonably good BLEU scores using one fifth of random samples:** For the multi-source multi-target case, we find that using one fifth of the low-resource training data gives reasonably good BLEU scores as shown in Figure 5. This is helpful when we have little low-resource data. For translation into low-resource language, the experts only need to translate a small amount of seed data before passing it to our system [1].

### 4.4 Order-preserving Lexiconized NMT

We devise a mechanism to build a parallel lexicon table across twenty-three European languages using very little data and zero manual work. A few lexicon examples are shown in Table 6. We first extract named entities from the English Bible (Manning et al., 2014) and combine them with English biblically named entities from multiple sources (Easton, 1897; Nave, 1903; Smith et al., 1967; Hitchcock, 1874; Rice, 2015). Secondly, we carefully automate the filtering process to obtain a clean English lexicon list. Using this list as the seed, we build a parallel lexicon table across all twenty-three languages through fast-aligning (Dyer et al., 2013). The final parallel lexicon table has 2916 named entities. In the translation task into low-resource language, we assume that the experts first translate these lexicon entries, and then translate approximately one fifth random sentences before we train our NMT. If necessary, the experts evaluate and correct translations before releasing the final translations to the low-resource language community. We aim to reduce human effort in post-editing and increase machine accuracy. After labeling named entities in each sentence pair in order, we train and obtain good translation results.

**We observe 60.6% accuracy in human evaluation where our translations are parallel to human translations:** In Table 8, we show some examples of machine translated text, we also show the expected correct translations for comparison. Not only the named entities are correctly mapped, but also the ordering of the subject and the object is preserved. In a subset of our test set, we conduct human evaluation on 320 English-Swedish results to rate the translations into three categories: accurate (parallel to human translation), almost accurate (needing minor corrections) and inaccurate. More precisely, each sentence is evaluated using three criteria: correct set of named entities, correct positioning of named entities, and accurate meaning of overall translation. If a sentence achieves all three, then it is termed as accurate; if either a name entity is missing or its position is wrong, then it is termed as almost accurate (needing minor cor-

[1] Note that using nine tenth of random samples yields higher performance than using full data, but it may not be generalized to other datasets.

| Source Sentence | NMT Translation without Order Preservation (Before) | NMT Translation with Order Preservation (After) | Correct Target Translation | Frequency of Named Entities |
|---|---|---|---|---|
| And *Noah* fathered three sons, *Shem*, *Ham*, and *Japheth*. | Och *Noa* födde tre söner, *Sem*, *Ham* och *Jafet*. | Och *Noa* födde tre söner, *Sem Ham* och *Jafet* | Och *Noa* födde tre söner: *Sem*, *Ham* och *Jafet*. | *Noah*: 58, *Shem*: 18, *Ham*: 17, *Japheth*: 11 |
| And *Saul* spoke to his son *Jonathan*, and to all his servants, to kill *David*. | Och *Saul* sade till *Jonatan*, hans son, och alla hans tjänare, så att de skulle döda *David*. | Och *Saul* talade till sin son *Jonatan* och alla hans tjänare för att döda *David* | Och *Saul* talade med sin son *Jonatan* och med alla sina tjänare om att döda *David* | *Saul*: 424, *Jonathan*: 121, *David*: 1134 |
| And they killed *Parshandatha*, and *Dalphon*, and *Aspatha*, and *Poratha*, and *Adalia*, and *Aridatha*, and *Parmashta*, and *Arisai*, and *Aridai*, and *Vajezatha*, | Och de dräpte *Kedak*, *Ir-Fittim*, *Aquila*, *dörrvaktarna*, *Amarja*, *Bered*, *vidare Bet-Hadt*, *Berota*, *Gat-Rimmon*, | Och de dräpte *Parsandata Dalefon* och *Aspata Porata Adalja Aridata Parmasta Arisai Aridai Vajsata* | Och *Parsandata*, *Dalefon*, *Aspata*, *Porata*, *Adalja*, *Aridata*, *Parmasta*, *Arisai*, *Aridai* och *Vajsata*, | *Parshandatha*: 1, *Dalphon*: 1, *Aspatha*: 1, *Poratha*: 1, *Adalia*: 1, *Aridatha*: 1, *Parmashta*: 1, *Arisai*: 1, *Aridai*: 1, *Vajezatha*: 1 |

Table 8: Examples of order-preserving lexicon-aware translation for English to Swedish. The frequency of the named entities are the number of occurrences each named entity appears in the whole dataset; for example, all named entities in the last sentence only appear in the test set once, and do not appear in the training data.

rection); if the meaning of the sentence is entirely wrong, then it is inaccurate. Our results are 60.6% accurate, 33.8% needing minor corrections, and 5.6% inaccurate. Though human evaluation carries bias and the sample is small, it does give us perspective on the performance of our model.

**Order-preservation performs well especially when the named entities are rare words:** In Table 8, NMT without order-preservation lexiconized treatment performs well when named entities are common words, but fails to predict the correct set of named entities and their ordering when named entities are rare words. The last column shows the number of occurrences of each named entity. For the last example, there are many named entities that only occur in data once, which means that they never appear in training and only appear in the test set. The normal NMT without order-preservation lexiconized treatment predicts the wrong set of named entities with the wrong ordering. Our lexiconized order-preserving NMT, on the contrary, performs well at both the head and tail of the distribution, predicts the right set of named entities with the right ordering.

**Prediction with longer sentences and many named entities are handled well:** In Table 8, we see that normal NMT without order-preservation lexiconized treatment performs well with short sentences and few named entities in a sentence. But as the number of the name entities per sentence increases, especially when the name entities are rare unknowns as discussed before, normal NMT cannot make correct prediction of the right set of name entities with the correct ordering 8. Our lexiconized order-preserving NMT, on the contrary, gives very high accuracy when there are many named entities in the sentence and maintains their correct ordering.

**Trimming the lexicon list that keeps the tail helps to increase BLEU scores:** Different from most of the previous lexiconized NMT works where BLEU scores never increase (Wang et al., 2017), our BLEU scores show minor improvements. BLEU score for German-Swedish translation increases from 35.8 to 36.6 in Table 7. As an attempt to increase our BLEU scores even further, we conduct two more experiments. In one setting, we keep only the tail of the lexicon table that occur in the Bible once. In another setting, we keep only a manual selection of lexicons. Note that this is the only place where manual work is involved and is not essential. There are minor improvements in BLEU scores in both cases.

**33.8% of the translations require minor corrections:** The sentence length for these translations that require minor corrections is often longer. We notice that some have repetitions that do not affect meaning, but need to be trimmed. Some have the under-prediction problem where certain named entities in the source sentence never appear; in this case, missing named entities need to be added. Some have minor issues with plurality and tense. We show a few examples of the translations that need minor corrections in the appendices for reference. Typically, sentences with longer sentence length and more complicated named entity relationships require minor corrections to achieve high translation quality.

## 5 Conclusion and Future Directions

We present our order-preserving translation system for cross-lingual learning in European languages. We examine three issues that are important to translation into low-resource language: the lack of low-resource data, effective cross-lingual transfer, and the variable-binding problem.

Firstly, we add the source and the target family labels in training and examined intra-family and inter-family effects. We find that training on multiple families, more specifically, training on two neighboring families nearest to the low-resource language improves BLEU scores to a reasonably good level. Secondly, we devise a rigorous ablation study and show that we only need a small portion of the low-resource target data to produce reasonably good BLEU scores. Thirdly, to address the variable-binding problem, we build a parallel lexicon table across twenty-three European languages and design a novel order-preserving named entity translation method by tagging named entities in each sentence in order. We achieve reasonably good quantitative and qualitative improvements in a preliminary study.

The order-preserving named entity translation labels named entities in order. Since there are relatively less number of long sentences with many named entities than short sentences with few named entities, underprediction of named entities in long sentences may occur. To seek solution to the underprediction problem, we are looking at randomized labeling of the named entities. Moreover, our order-preserving named entity translation method works well with a fixed pool of named entities in any static document known in advance. This is due to our unique use cases for applications like translating water, sanitation and hygiene (WASH) guidelines written in the introduction. We devise our method to ensure high accuracy targeting translating named entities in static document known in advance. However, researchers may need to translate dynamic document to low-resource language in real-time. We are actively researching into the dynamic timely named entity discovery with high accuracy.

We are actively extending our work to cover more world languages, more diverse domains, and more varied sets of datasets to show our methods are generalizable. Since our experiments shown in this paper are using European languages, we are also interested on non-European languages like Arabic, Indian, Chinese, Indonesian and many others to show that our model is widely generalizable. We also expect to discover interesting research ideas exploring a wider universe of linguistically dissimilar languages.

Our work is helpful for translation into low-resource language, where human translators only need to translate a few lexicons and a partial set of data before passing it to our system. Human translators may also be needed during post-editing before a fully accurate translation is released. Our future goal is to minimize the human correction efforts and to present high quality translation timely.

We would also like to work on real world low-resource tribal languages where there is no or little training data. Translation using limited resources and data in these tribal groups that fits with the culture-specific rules will be very important (Levin et al., 1998). Real world low-resource languages call for cultural-aware translation.

## Acknowledgments

We would like to thank Prof. Eduard Hovy for his insights on the topic and helpful suggestions. We would also like to thank Prof. Michael Cysouw for his generous sharing of the massive Bible corpus.

## References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the 17th Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Ulrich Ammon. 2001. *The dominance of English as a language of science: Effects on other languages and language communities*, volume 84. Walter de Gruyter.

Dimitra Anastasiou and Reinhard Schäler. 2010. Translating vital information: Localisation, internationalisation, and globalisation. *Syn-thèses Journal*, 3:11–25.

Antonios Anastasopoulos, Sameer Bansal, David Chiang, Sharon Goldwater, and Adam Lopez. 2017. Spoken term discovery for language documentation using translations. In *Proceedings of the Workshop on Speech-Centric Natural Language Processing*, pages 53–58.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 21st Conference on Empirical Methods in Natural Language Processing*.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 16th Conference on Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics.

Rafael E Banchs and Marta R Costa-Jussà. 2011. A semantic feature for statistical machine translation. In *Proceedings of the 5th workshop on syntax, semantics and structure in statistical translation*, pages 126–134. Association for Computational Linguistics.

Julia Barrett. 2005. Support and information needs of older and disabled older people in the uk. *Applied ergonomics*, 36(2):177–183.

Stephen Beale, Sergei Nirenburg, Marjorie McShane, and Tod Allman. 2005. Document authoring the bible for minority language translation. *Proceedings of MT-Summit, Phuket, Thailand.*

Jasone Cenoz. 2001. The effect of linguistic distance, l2 status and age on cross-linguistic influence in third language acquisition. *Cross-linguistic influence in 2nd language acquisition: Psycholinguistic perspectives*, 111(45):8–20.

Sin-wai Chan and David E Pollard. 2001. *An Encyclopaedia of Translation: Chinese-English, English-Chinese*. Chinese University Press.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language resources and evaluation*, 49(2):375–395.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1693–1703.

Boele De Raad, Marco Perugini, and Zsófia Szirmák. 1997. In pursuit of a cross-lingual reference structure of personality traits: Comparisons among five languages. *European Journal of Personality*, 11(3):167–185.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1723–1732.

Philipp Dufter and Hinrich Schütze. 2018. A universal semantic space. *arXiv preprint arXiv:1801.06807*.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. An attentional model for speech translation without transcription. In *Proceedings of the 15th Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies*, pages 949–959.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2017. Multilingual training of crosslingual word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 894–904.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 12th Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies*, pages 644–648.

Matthew George Easton. 1897. *Eastons Bible Dictionary: A Dictionary of Bible Terms*. Thomas Nelson.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Low cost portability for statistical machine translation based on n-gram frequency and tf-idf. In *International Workshop on Spoken Language Translation*.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 15th Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies*, pages 866–875.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of the 15th Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies*, pages 1296–1306.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Rosalind M Harding and Robert R Sokal. 1988. Classification of the european language families by genetic distance. *Proceedings of the National Academy of Sciences*, 85(23):9370–9372.

Theo Hermans. 2003. Cross-cultural translation studies as thick translation. *Bulletin of the School of Oriental and African Studies*, 66(3):380–389.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. 2017. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*.

RD Hitchcock. 1874. Hitchcock's bible names dictionary, art. *AJ Johnson Publishers, New York.*

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *Proceedings of the 55th annual meeting of the Association for Computational Linguistics, System Demonstrations*, pages 67–72.

Lori Levin, Donna Gates, Alon Lavie, and Alex Waibel. 1998. An interlingua based on domain actions for machine translation of task-oriented dialogues. In *Proceedings of the 5th International Conference on Spoken Language Processing*.

Jindřich Libovickỳ and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 196–202.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics*, pages 55–60.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel bible corpus. *Oceania*, 135(273):40.

Martijn Naaijer and Dirk Roorda. 1993. Parallel texts in the hebrew bible, new methods and visualizations. *Young*, 140:157.

Orville James Nave. 1903. *Nave's Topical Bible: A Digest of the Holy Scriptures*. Topical Bible Publishing Company.

Toan Q Nguyen and David Chiang. 2017. Improving lexical choice in neural machine translation. *arXiv preprint arXiv:1710.01329*.

Terence Odlin. 1989. *Language transfer: Cross-linguistic influence in language learning*. Cambridge University Press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Robyn Perry and Steven Bird. 2017. Treasure language storytelling: Cross-cultural language recognition and wellbeing. *Proceedings of the 5th International Conference on Language Documentation and Conservation.*

Filippo Petroni and Maurizio Serva. 2008. Language distance and tree reconstruction. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(08):P08012.

Lutz Prechelt. 1998. Early stopping-but when? *Neural Networks: Tricks of the trade*, pages 553–553.

B Reddy, Yadlapalli S Kusuma, Chandrakant S Pandav, Anil Kumar Goswami, Anand Krishnan, et al. 2017. Water and sanitation hygiene practices for under-five children among households of sugali tribe of chittoor district, andhra pradesh, india. *Journal of environmental and public health.*

Philip Resnik, Mari Broman Olsen, and Mona Diab. 1999. The bible as a parallel corpus: Annotating the 'book of 2000 tongues'. *Computers and the Humanities*, 33(1-2):129–153.

Edwin W. Rice. 2015. *People's Dictionary of the Bible*. Forgotten Books.

Malcolm Ross et al. 2006. Language families and linguistic diversity. In *Encyclopedia of Language and Linguistics*, 2 edition. Elsevier.

Edward Sapir. 1921. How languages influence each other. *Language: an Introduction to the Study of Speech.*

Kevin P Scannell. 2006. Machine translation for closely related language pairs. In *Proceedings of the Workshop Strategies for developing machine translation for minority languages*, pages 103–109. Citeseer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Philippa Shoemark, Sharon Goldwater, James Kirby, and Rik Sarkar. 2016. Towards robust cross-linguistic comparisons of phonological networks. In *Proceedings of the 14th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 110–120.

William Smith, Francis Nathan Peloubet, and Mary Abby Thaxter Peloubet. 1967. *Smith's Bible Dictionary*. Pyramid Books.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multi-modal machine translation and crosslingual image description. In *Proceedings of the 1st Conference on Machine Translation*, volume 2, pages 543–553.

Jörg Tiedemann. 2012. Character-based pivot translation for under-resourced languages and domains. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 141–151. Association for Computational Linguistics.

Antonio Toral and Andy Way. 2018. What level of quality can neural machine translation attain on literary text? *arXiv preprint arXiv:1801.04962*.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *Proceedings of the 15th Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies*, pages 1357–1366.

Marco Turchi, Tijl De Bie, and Nello Cristianini. 2008. Learning performance of a machine translation system: a statistical and computational analysis. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 35–43. Association for Computational Linguistics.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017. Sogou neural machine translation systems for wmt17. In *Proceedings of the 2nd Conference on Machine Translation*, pages 410–415.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 15th Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies*, pages 30–34.

# Trivial Transfer Learning for Low-Resource Neural Machine Translation

**Tom Kocmi** **Ondřej Bojar**

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
`<surname>@ufal.mff.cuni.cz`

## Abstract

Transfer learning has been proven as an effective technique for neural machine translation under low-resource conditions. Existing methods require a common target language, language relatedness, or specific training tricks and regimes. We present a simple transfer learning method, where we first train a "parent" model for a high-resource language pair and then continue the training on a low-resource pair only by replacing the training corpus. This "child" model performs significantly better than the baseline trained for low-resource pair only. We are the first to show this for targeting different languages, and we observe the improvements even for unrelated languages with different alphabets.

## 1 Introduction

Neural machine translation (NMT) has made a big leap in performance and became the unquestionable winning approach in the past few years (Bahdanau et al., 2014; Sutskever et al., 2014; Sennrich et al., 2017; Vaswani et al., 2017). The main reason behind the success of NMT in realistic conditions was the ability to handle large vocabulary (Sennrich et al., 2016b) and to utilize large monolingual data (Sennrich et al., 2016a). However, NMT still struggles if the parallel data is insufficient (e.g. fewer than 1M parallel sentences), producing fluent output unrelated to the source and performing much worse than phrase-based machine translation (Koehn and Knowles, 2017).

Many strategies have been used in MT in the past for employing resources from additional languages, see e.g. Wu and Wang (2007), Nakov and Ng (2012), El Kholy et al. (2013), or Hoang and Bojar (2016). For NMT, a particularly promising approach is transfer learning or "domain adaptation" where the "domains" are the different languages.

For example, Zoph et al. (2016) train a "parent" model in a high-resource language pair, then use some of the trained weights as the initialization for a "child" model and further train it on the low-resource language pair. In Zoph et al. (2016), the parent and child pairs shared the target language (English) and a number of modifications of the training process were needed to achieve an improvement in translation from Hansa, Turkish, and Uzbek into English with the help of French-English data.

Nguyen and Chiang (2017) explore a related scenario where the parent language pair is also low-resource but it is related to the child language pair. They improved the previous approach by using a shared vocabulary of subword units (BPE, Sennrich et al., 2016b). Additionally, they used transliteration to improve their results.

In this paper, we contribute empirical evidence that transfer learning for NMT can be simplified even further. We leave out the restriction on relatedness of the languages and extend the experiments to parent–child pairs where the target language changes. Moreover, we do not utilize any special modifications to the training regime or data pre-preprocessing.

In contrast to previous work, we test the method with the Transformer model (Vaswani et al., 2017), instead of the recurrent approaches (Bahdanau et al., 2014). As documented in e.g. Popel and Bojar (2018) and anticipated in WMT18,[1] the Transformer model seems superior to other NMT approaches.

## 2 Method Description

The proposed method is extremely simple: We train the parent language pair for a number of iter-

[1] `http://www.statmt.org/wmt18/translation-task.html`

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 244–252
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64025

ations and switch the training corpus to the child language pair for the rest of the training, without resetting any of the training (hyper)parameters.

As such, this method is similar to the transfer learning proposed by Zoph et al. (2016) but uses the shared vocabulary as in Nguyen and Chiang (2017). The novelty is that we are removing the restriction about relatedness of the language pairs, and in contrast to the previous papers, we show that this simple style of transfer learning can be used on both sides (i.e. either the source or the target language), not only with the target language common to both parent and child model. In fact, the method is effective also for fully unrelated language pairs.

Our method does not need any modification of existing NMT frameworks. The only requirement is to use a shared vocabulary of subword units (we use wordpieces, Johnson et al., 2017) across both language pairs. This is achieved by learning wordpiece segmentation from the concatenated source and target sides of both the parent and child language pairs. All other parameters of the model stay the same as for the standard NMT training.

During the training we first train the NMT model for the high-resource language pair until convergence. This model is called "parent". After that, we train the child model without any restart, i.e. only by changing the training corpora to the low-resource language pair.

### 2.1 Details on Shared Vocabulary

Current NMT systems use vocabularies of subword units instead of whole words. Using subword units gives a balance between the flexibility of separate characters and efficiency of whole words. It solves the out-of-vocabulary words problem and reduces the vocabulary size. The majority of NMT systems use either the byte pair encoding (Sennrich et al., 2016b) or wordpieces (Wu et al., 2016). Given a training corpus and the desired maximal vocabulary size, either method produces deterministic rules for word segmentation to achieve the fewest possible splits.

Our method requires the vocabulary shared across both the parent (translating from language XX to YY) and the child model (translating from AA to BB). This is obtained by concatenating both training corpora into one corpus of sentences in languages AA, BB, XX and YY. [2]

Due to our focus on low-resource language pairs, we decided to generate the vocabulary in a balanced way by selecting the same amount of sentences from both language pairs. We thus use the same number of sentence pairs of the parent corpus as there are in the child corpus.

We did not experiment with any other balancing of the vocabulary. Future research could also investigate the impact of using only the child corpus for vocabulary generation or various amounts of used sentences.

We generated vocabularies aiming at 32k subword types. The exact size of the vocabulary varies from 26.1k to 34.8k. All experiments of a given language set use the same vocabulary. Vocabulary overlap in each language set is further studied in Section 6.1.

## 3 Model Description

We use the Transformer sequence-to-sequence model (Vaswani et al., 2017) as implemented in Tensor2Tensor (Vaswani et al., 2018) version 1.4.2. Our models are based on the "big single GPU" configuration as defined in the paper. To fit the model to our GPUs (NVIDIA GeForce GTX 1080 Ti with 11 GB RAM), we set the batch size to 2300 tokens and limit sentence length to 100 wordpieces.

We use exponential learning rate decay with the starting learning rate of 0.2 and 32000 warm up steps and Adam optimized. In our experiments, we find that it is undesirable to reset learning rate as it leads to the loss of the performance from the parent model. Therefore the transfer learning is handled only by changing the training corpora and nothing else.

Decoding uses the beam size of 8 and the length normalization penalty is set to 1.

The models were trained for 1M steps (approx. 140 hours), which was sufficient for models to converge to the best performance. We selected the model with the best performance on the development test for the final evaluation on the testset.

## 4 Datasets

In our experiments, we compare low-resource and high-resource language pairs spanning two orders

[2] Having separate vocabularies for the parent and child and switching from the XX-YY to AA-BB vocabulary when we switch the training corpus leads on an expected drop in performance. Independent vocabularies use different IDs even for identical subwords and the network cannot rely on any of its weights from the parent training.

| Lang. | Sent. | Words | | Vocabulary | |
|---|---|---|---|---|---|
| pair | pairs | First | Second | First | Second |
| ET,EN | 0.8 M | 14 M | 20 M | 631 k | 220 k |
| FI,EN | 2.8 M | 44 M | 64 M | 1697 k | 545 k |
| SK,EN | 4.3 M | 82 M | 95 M | 1059 k | 610 k |
| RU,EN | 12.6 M | 297 M | 321 M | 2202 k | 3161 k |
| CS,EN | 40.1 M | 491 M | 563 M | 6253 k | 4130 k |
| AR,RU | 10.2 M | 243 M | 252 M | 2299 k | 2099 k |
| FR,RU | 10.0 M | 295 M | 238 M | 1339 k | 2045 k |
| ES,FR | 10.0 M | 297 M | 288 M | 1426 k | 1323 k |
| ES,RU | 10.0 M | 300 M | 235 M | 1433 k | 2032 k |

Table 1: Datasets sizes overview. We consider Estonian and Slovak low-resource languages in our paper. Word counts and vocabulary sizes are from the original corpus, tokenizing only at whitespace and preserving the case.

of magnitude of training data sizes. We consider Estonian (ET) and Slovak (SK) as low-resource languages compared to the Finnish (FI) and Czech (CS) counterparts.

The choice of languages was closely related to the languages in this year's WMT 2018 shared tasks. In particular, Estonian and Finnish (paired with English) were suggested as the main focus for their relatedness. We added Czech and Slovak as another closely related language pair. Russian (RU) for the parent model was chosen for two reasons: (1) written in Cyrillic, there will be hardly any intersection in the shared vocabulary with the child language pairs, and (2) previous work uses transliteration to handle Russian, which is a nice contrast to our work. Finally, we added Arabic (AR), French (FR) and Spanish (ES) for experiments with unrelated languages.

The sizes of the training datasets are in Table 1.

If not specified otherwise we use training, development and test sets from WMT.[3] Pairs with training sentences with less than 4 words or more than 75 words on either the source or the target side are removed to allow for a speedup of Transformer by capping the maximal length and allowing a bigger batch size. The reduction of training data is small and based on our experiments, it does not change the performance of the translation model.

We use the Europarl and Rapid corpora for Estonian-English. We disregard Paracrawl due to its noisiness. The development and test sets are from WMT news 2018.

The Finnish-English was prepared as in Östling et al. (2017), removing Wikipedia headlines. The dev and test sets are from WMT news 2015.

For English-Czech, we use all paralel data allowed in WMT2018 except Paracrawl. The main resource is CzEng 1.7 (the filtered version, Bojar et al., 2016). The devset is WMT newstest2011 and the testset is WMT newstest2017.

Slovak-English uses corpora from Galušcáková and Bojar (2012), detokenized by Moses.[4] WMT newstest2011 serves as the devset and testset.

The Russian-English training set was created from News Commentary, Yandex and UN Corpus. As the devset, we use WMT newstest 2012.

The language pairs Arabic-Russian, French-Russian, Spanish-French and Spanish-Russian were selected from UN corpus (Ziemski et al., 2016), which provides over 10 million multi-parallel sentences in 6 languages.

## 5 Results

In this section, we present results of our approach. Statistical significance of the winner (marked with ‡) is tested by paired bootstrap resampling against the baseline (child-only) setup (1000 samples, conf. level 0.05; Koehn, 2004).

As customary, we label the models with the pair of the source and target language codes, for example the English-to-Estonian translation model is denoted by ENET.

The vocabularies are generated as described in 2.1 separately for each experimented combination of parent and child. The same vocabulary is used whenever the parent and child use the same set of languages, i.e. disregarding the translation direction and model stage (parent or child).

### 5.1 English as the Common Language

Table 2 summarizes our results for various combinations of high-resource parent and low-resource child language pairs when English is shared between the child and parent either in the encoder or in the decoder.

We confirm that sharing the target language improves performance as previously shown (Zoph et al., 2016; Nguyen and Chiang, 2017). This gains up to 2.44 BLEU absolute for ETEN with the FIEN parent. Using only the parent (FIEN) model to translate the child (ETEN) test set gives a miserable performance, confirming the need for transfer learning or "finetuning".

[3] http://www.statmt.org/wmt18/

[4] https://github.com/moses-smt/mosesdecoder

|  |  | Baselines: Only |  |
|---|---|---|---|
| Parent - Child | Transfer | Child | Parent |
| enFI - enET | 19.74‡ | 17.03 | 2.32 |
| FIen - ETen | 24.18‡ | 21.74 | 2.44 |
| **enCS - enET** | 20.41‡ | 17.03 | 1.42 |
| **enRU - enET** | 20.09‡ | 17.03 | 0.57 |
| **RUen - ETen** | 23.54‡ | 21.74 | 0.80 |
| enCS - enSK | 17.75‡ | 16.13 | 6.51 |
| CSen - SKen | 22.42‡ | 19.19 | 11.62 |
| enET - enFI | 20.07‡ | 19.50 | 1.81 |
| ETen - FIen | 23.95 | 24.40 | 1.78 |
| enSK - enCS | 22.99 | 23.48‡ | 6.10 |
| SKen - CSen | 28.20 | 29.61‡ | 4.16 |

Table 2: Transfer learning with English reused either in source (encoder) or target (decoder). The column "Transfer" is our method, baselines correspond to training on one of the corpora only. Scores (BLEU) are always for the child language pair and they are comparable only within lines or when the child language pair is the same. "Unrelated" language pairs in bold. Upper part: parent larger, lower part: child larger. ("EN" lowercased just to stand out.)

A novel result is that the method works also for sharing the source language, improving ENET by up to 2.71 BLEU thanks to ENFI parent.

Furthermore, the improvement is not restricted only to related languages as Estonian and Finnish as shown in previous works. Unrelated language pairs (shown in bold in Table 2) like Czech and Estonian work too and in some cases even better than with the related datasets. We reach an improvement of 3.38 BLEU for ENET when parent model was ENCS, compared to improvement of 2.71 from ENFI parent. This statistically significant improvement contradicts Dabre et al. (2017) who concluded that the more related the languages are, the better transfer learning works. We see it as an indication that the size of the parent training set is more important than relatedness of languages.

The results with Russian parent for Estonian child (both directions) show that transliteration is also not necessary. Because there is no vocabulary sharing between Russian Cyrilic and Estonian Latin (except numbers and punctuation, see Section 6.1 for further details), the improvement could be attributed to a better coverage of English; an effect similar to domain adaptation.

On the other hand, this transfer learning works well only when the parent has more training data than the child. As presented in the bottom part of Table 2, low-resource parents do not generally improve the performance of better-resourced childs and sometimes, they even (significantly) decrease it. This is another indication, that the most important is the size of the parent corpus compared to the child one.

| Child Training Sents | Transfer BLEU | Baseline BLEU |
|---|---|---|
| 800k | 19.74 | 17.03 |
| 400k | 19.04 | 14.94 |
| 200k | 17.95 | 11.96 |
| 100k | 17.61 | 9.39 |
| 50k | 15.95 | 5.74 |
| 10k | 12.46 | 1.95 |

Table 3: Maximal score reached by ENET child for decreasing sizes of child training data, trained off an ENFI parent (all ENFI data are used and models are trained for 800k steps). The baselines use only the reduced ENET data.

The baselines are either models trained purely on the child parallel data or only on the parent data. The second baseline only indicates the relatedness of languages because it is only tested but never trained on the child language pair. Also, we do not add any language tag as in Johnson et al. (2017). This also highlights that the improvement of our method cannot be directly attributed to the relatedness of languages: e.g. Czech and Slovak are much more similar than Czech and Estonian (Parent Only BLEU of translation out of English is 6.51 compared to 1.42) and yet the gain from transfer learning is larger for Estonian (+3.38) than from Slovak (+1.62).

### 5.2 Simulated Very Low Resources

In Table 3, we simulate very low-resource settings by downscaling the data for the child model. It is a common knowledge, that gains from transfer learning are more pronounced for smaller childs. The point of Table 3 is to illustrate that our approach is applicable even to extremely small child setups, with as few as 10k sentence pairs. Our transfer learning ("start with a model for whatever parent pair") may thus resolve the issue of applicability of NMT for low resource languages as pointed out by Koehn and Knowles (2017).

### 5.3 Parent Convergence

Figure 1 compares the performance of the child model when trained from various training stages of the parent model. The performance of the child clearly correlates with the performance of the parent. Therefore, it is better to use a parent model that already converged and reached its best performance.

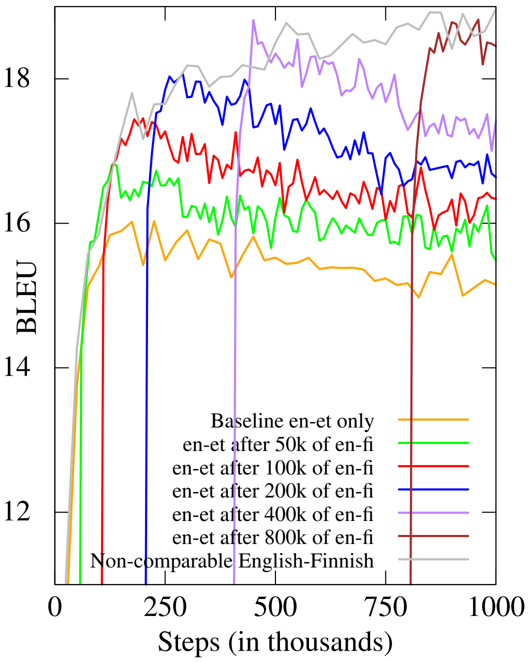


Figure 1: Learning curves on dev set for ENFI parent and ENET child where the child model started training after various numbers of the parent's training steps.

| Parent - Child | Transfer | Baseline | Aligned |
|---|---|---|---|
| enFI - ETen | 22.75‡ | 21.74 | 24.18 |
| FIen - enET | 18.19‡ | 17.03 | 19.74 |
| enRU - ETen | 23.12‡ | 21.74 | 23.54 |
| enCS - ETen | 22.80‡ | 21.74 | not run |
| RUen - enET | 18.16‡ | 17.03 | 20.09 |
| enET - ETen | 22.04‡ | 21.74 | 21.74 |
| ETen - enET | 17.46 | 17.03 | 17.03 |

Table 4: Results of child following a parent with swapped direction. "Baseline" is child-only training. "Aligned" is the more natural setup with English appearing on the "correct" side of the parent, the numbers in this column thus correspond to those in Table 2.

### 5.4 Direction Swap in Parent and Child

Relaxing the setup in Section 5.1, we now allow a mismatch in translation direction of the parent and child. The parent XX-EN is thus followed by an EN-YY child or vice versa. It is important to note that Transformer shares word embeddings for the source and target side. The gain can be thus due to better English word embeddings, but definitely not due to a better English language model. It would be interesting to study the effect of not sharing the embeddings but we leave it for some future work.

The results in Table 4 document that an improvement can be reached even when none of the involved languages is reused on the same side. This interesting result should be studied in more detail. Firat et al. (2016) hinted possible gains even when both languages are distinct from the low-resource languages but in a multilingual setting. Not surprisingly, the improvements are better when the common language is aligned.

The bottom part of Table 4 shows a particularly interesting trick: the parent is not any high-resource pair but the very same EN-ET corpus with source and target swapped. We see gains in both directions, although not always statistically significant. Future work should investigate if this performance boost is possible even for high-resource languages. Similar behavior has been shown in Niu et al. (2018), where in contrast to our work they mixed the data together and added an artificial token indicating the target language.

| Parent - Child | Transfer | Baseline |
|---|---|---|
| ARRU - ETEN | 22.23 | 21.74 |
| ESFR - ETEN | 22.24‡ | 21.74 |
| ESRU - ETEN | 22.52‡ | 21.74 |
| FRRU - ETEN | 22.40‡ | 21.74 |

Table 5: Transfer learning with parent and child not sharing any language.

### 5.5 No Language in Common

Our final set of experiments examines the performance of ETEN child trained off parents in totally unrelated language pairs. Without any common language, the gains cannot be attributed, e.g., to the shared English word embeddings. The vocabulary overlap is mostly due to short n-grams or numbers and punctuations.

We see gains from transfer learning in all cases, mostly significant. The only non-significant gain is from Arabic-Russian which does not share the script with the child Latin at all. (Sharing of punctuation and numbers is possible across all the tested scripts.) The gains are quite similar (+0.49–+0.78 BLEU), supporting our assumption that the main factor is the size of the parent (here, all have 10M sentence pairs) rather than language relatedness.

## 6 Analysis

Here we provide a rather initial analysis of the sources of the gains.

### 6.1 Vocabulary Overlap

Out method relies on the vocabulary estimated jointly from the child and parent model. In Trans-

| ET | EN | RU | % Subwords |
|---|---|---|---|
| ✓ | - | - | 29.93% |
| - | ✓ | - | 20.69% |
| - | - | ✓ | 29.03% |
| ✓ | ✓ | - | 10.06% |
| - | ✓ | ✓ | 1.39% |
| ✓ | - | ✓ | 0.00% |
| ✓ | ✓ | ✓ | 8.89% |
| Total | | | 28.2k (100%) |
| From parent | | | 41.03% |

Table 6: Breakdown of subword vocabulary of experiments involving ET, EN and RU.

former, the vocabulary is even shared across encoder and decoder. With a large overlap, we could expect a lot of "information reuse" between the parent and the child.

Since the subword vocabulary depends on the training corpora, a little clarification is needed. We take the vocabulary of subword units as created e.g. for ENRU-ENET experiments, see Section 2.1. This vocabulary contains 28.2k subwords in total. We then process the training corpora for each of the languages with this shared vocabulary, ignore all subwords that appear less than 10 times in each of the languages (these subwords will have little to no impact on the result of the training) and break down the total 28.2k subwords into classes depending on the languages in which the particular subword was observed, see Table 6.

We see that the vocabulary is reasonably balanced, with each language having 20–30% of subwords unique to it. English and Estonian share 10% subwords not seen in Russian while Russian shares only 0–1.39% of subwords with each of the other languages. Overall 8.89% of subwords are seen in all three languages.

A particularly interesting subset is the one where parent languages help the child model, in other words subwords appearing anywhere in English and also tokens common to Estonian and Russian. For this set of languages, this amounts to 20.69+10.06+1.39+0.0+8.89 = 41.03%. We list this number on a separate line in Table 6, "From parent". These subwords get their embeddings trained better thanks to the parent model.

Table 7 summarizes this analysis for several language sets, listing what portion of subwords is unique to individual languages in the set, what portion is shared by all the languages and what portion of subwords benefits from the parent training. We see a similar picture across the board, only AR-RU-ET-EN stands out with the very low number of subwords (6.2%) available already in the parent. The parent AR-RU thus offered very little word knowledge to the child and yet lead to a gain in BLEU.

| Languages | Unique in a Lang. | In All | From Parent |
|---|---|---|---|
| ET-EN-FI | 24.4-18.2-26.2 | 19.5 | 49.4 |
| ET-EN-RU | 29.9-20.7-29.0 | 8.9 | 41.0 |
| ET-EN-CS | 29.6-17.5-21.2 | 20.3 | 49.2 |
| AR-RU-ET-EN | 28.6-27.7-21.2-9.1 | 4.6 | 6.2 |
| ES-FR-ET-EN | 15.7-13.0-24.8-8.8 | 18.4 | 34.1 |
| ES-RU-ET-EN | 14.7-31.1-21.3-9.3 | 6.0 | 21.4 |
| FR-RU-ET-EN | 12.3-32.0-22.3-8.1 | 6.3 | 23.1 |

Table 7: Summary of vocabulary overlaps for the various language sets. All figures in % of the shared vocabulary.

| | BLEU | nPER | nTER | nCDER | chrF3 | nCharacTER |
|---|---|---|---|---|---|---|
| Base ENET | 16.13 | 47.13 | 32.45 | 36.41 | 48.38 | 33.23 |
| ENRU+ENET | 19.10 | 50.87 | 36.10 | 39.77 | 52.12 | 39.39 |
| ENCS+ENET | 19.30 | 51.51 | 36.84 | 40.42 | 52.71 | 40.81 |

Table 8: Various automatic scores on ENET test set. Scores prefixed "n" reported as $(1 - \text{score})$ to make higher numbers better.

### 6.2 Output Analysis

Since we rely on automatic analysis, we need to prevent some potential overestimations of translation quality due to BLEU. For this, we took a closer look at the baseline ENET model (BLEU of 17.03 in Table 2) and two ENET childs derived from ENCS (BLEU of 20.41) and ENRU parent (BLEU 20.09).

Table 8 confirms the improvements are not an artifact of uncased BLEU. The gains are apparent with several (now cased) automatic scores.

As documented in Table 9, the improved outputs are considerably longer. In the table, we show also individual $n$-gram precisions and brevity penalty (BP) of BLEU. The longer output clearly helps to reduce the incurred BP but the improvements are also apparent in $n$-gram precisions. In other words, the observed gain cannot be attributed solely to producing longer outputs.

Table 10 explains the gains in unigram precisions by checking which tokens in the improved outputs (the parent followed by the child) were present also in the baseline (child-only, denoted "b" in Table 10) and/or confirmed by the reference (denoted "r"). We see that about 44+20% of tokens of improved outputs can be seen as "unchanged" compared to the baseline because they appear already in the baseline output ("b"). (The

|  | Length | BLEU Components | BP |
|---|---|---|---|
| Base ENET | 35326 | 48.1/21.3/11.3/6.4 | 0.979 |
| ENRU+ENET | 35979 | 51.0/24.2/13.5/8.0 | 0.998 |
| ENCS+ENET | 35921 | 51.7/24.6/13.7/8.1 | 0.996 |

Table 9: Candidate total length, BLEU $n$-gram precisions and brevity penalty (BP). The reference length in the matching tokenization was 36062.

|  | ENRU+ENET | ENCS+ENET |
|---|---|---|
| rb | 15902 (44.2 %) | 15924 (44.3 %) |
| - | 9635 (26.8 %) | 9485 (26.4 %) |
| b | 7209 (20.0 %) | 7034 (19.6 %) |
| r | 3233 (9.0 %) | 3478 (9.7 %) |
| Total | 35979 (100.0 %) | 35921 (100.0 %) |

Table 10: Comparison of improved outputs vs. the baseline and reference.

44% “rb” tokens are actually confirmed by the reference.)

The differing tokens are more interesting: “-” denotes the cases when the improved system produced something different from the baseline and also from the reference. Gains in BLEU are due to “r” tokens, i.e. tokens only in the improved outputs and the reference but not the baseline “b”. For both parent setups, there are about 9–9.7 % of such tokens. We looked at these 3.2k and 3.5k tokens and we have to conclude that these are regular *Estonian* words; no Czech or Russian leaks to the output and the gains are *not* due to simple token types common to all the languages (punctuation, numbers or named entities). We see identical BLEU gains even if we remove all such simple tokens from the candidates and references. A better explanation of the gains thus still has to be sought for.

## 7 Related Work

Firat et al. (2016) propose multi-way multi-lingual systems, with the main goal of reducing the total number of parameters needed to cater multiple source and target languages. To keep all the language pairs “active” in the model, a special training schedule is needed. Otherwise, catastrophic forgetting would remove the ability to translate among the languages trained earlier.

Johnson et al. (2017) is another multi-lingual approach: all translation pairs are simply used at once and the desired target language is indicated with a special token at the end of the source side. The model implicitly learns translation between many languages and it can even translate among language pairs never seen together.

Lack of parallel data can be tackled by unsupervised translation (Artetxe et al., 2018; Lample et al., 2018). The general idea is to mix monolingual training of autoencoders for the source and target languages with translation trained on data translated by the previous iteration of the system.

When no parallel data are available, the trainset of closely related high-resource pair can be used with transliteration approach as described in Karakanta et al. (2018).

Aside from the common back-translation (Sennrich et al., 2016a; Kocmi et al., 2018), simple copying of target monolingual data back to source (Currey et al., 2017) has been also shown to improve translation quality in low-data conditions.

Similar to transfer learning is also curriculum learning (Bengio et al., 2009; Kocmi and Bojar, 2017), where the training data are ordered from foreign out-of-domain to the in-domain training examples.

## 8 Conclusion

We presented a simple method for transfer learning in neural machine translation based on training a parent high-resource pair followed a low-resource language pair dataset. The method works for shared source or target side as well as for language pairs that do not share any of the translation sides. We observe gains also from totally unrelated language pairs, although not always significant.

One interesting trick we propose for low-resource languages is to start training in the opposite direction and swap to the main one afterwards.

The reasons for the gains are yet to be explained in detail but our observations indicate that the key factor is the size of the parent corpus rather than e.g. vocabulary overlaps.

## Acknowledgments

This study was supported in parts by the grants SVV 260 453, GAUK 8502/2016, and 18-24210S of the Czech Science Foundation. This work has been using language resources and tools stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (projects LM2015071 and OP VVV VI CZ.02.1.01/0.0/0.0/16 013/0001781).

## References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. Czeng 1.6: enlarged czech-english parallel corpus with processing tools dockered. In *International Conference on Text, Speech, and Dialogue*, pages 231–238. Springer.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156.

Raj Dabre, Tetsuji Nakagawa, and Hideto Kazawa. 2017. An empirical study of language relatedness for transfer learning in neural machine translation. In *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, pages 282–286.

Ahmed El Kholy, Nizar Habash, Gregor Leusch, Evgeny Matusov, and Hassan Sawaf. 2013. Language independent connectivity strength features for phrase pivot statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Sofia, Bulgaria. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Petra Galušcáková and Ondrej Bojar. 2012. Improving smt by using parallel data of a closely related language. In *Proc. of HLT*, pages 58–65.

Duc Tam Hoang and Ondrej Bojar. 2016. Pivoting methods and data for czech-vietnamese translation via english. *Baltic Journal of Modern Computing*, 4(2):190–202.

Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernand a Vigas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Alina Karakanta, Jon Dehdari, and Josef van Genabith. 2018. Neural machine translation for low-resource languages without parallel corpora. *Machine Translation*, 32(1):167–189.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum Learning and Minibatch Bucketing in Neural Machine Translation. In *Recent Advances in Natural Language Processing 2017*.

Tom Kocmi, oman Sudarikov, and Ondřej Bojar. 2018. CUNI Submissions in WMT18. In *Proceedings of the 3rd Conference on Machine Translation (WMT)*, Brussels, Belgium.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, volume 4, pages 388–395.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Preslav Nakov and Hwee Tou Ng. 2012. Improving statistical machine translation for a resource-poor language using related resource-rich languages. *Journal of Artificial Intelligence Research*, 44:179–222.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301. Asian Federation of Natural Language Processing.

Xing Niu, Michael Denkowski, and Marine Carpuat. 2018. Bi-directional neural machine translation with synthetic parallel data. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 84–91, Melbourne, Australia. Association for Computational Linguistics.

Robert Östling, Yves Scherrer, Jörg Tiedemann, Gongbo Tang, and Tommi Nieminen. 2017. The helsinki neural machine translation system. In *Proceedings of the Second Conference on Machine Translation*, pages 338–347, Copenhagen, Denmark. Association for Computational Linguistics.

Martin Popel and Ondej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for Neural Machine Translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 193–199, Boston, MA. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Hua Wu and Haifeng Wang. 2007. Pivot language approach for phrase-based statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 856–863, Prague, Czech Republic. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *LREC*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

# Input Combination Strategies for Multi-Source Transformer Decoder

**Jindřich Libovický** and **Jindřich Helcl** and **David Mareček**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
{libovicky, helcl, marecek}@ufal.mff.cuni.cz

## Abstract

In multi-source sequence-to-sequence tasks, the attention mechanism can be modeled in several ways. This topic has been thoroughly studied on recurrent architectures. In this paper, we extend the previous work to the encoder-decoder attention in the Transformer architecture. We propose four different input combination strategies for the encoder-decoder attention: serial, parallel, flat, and hierarchical. We evaluate our methods on tasks of multimodal translation and translation with multiple source languages. The experiments show that the models are able to use multiple sources and improve over single source baselines.

## 1 Introduction

The Transformer model (Vaswani et al., 2017) recently demonstrated superior performance in neural machine translation (NMT) and other sequence generation tasks such as text summarization or image captioning (Kaiser et al., 2017). However, all of these setups consider only a single input to the decoder part of the model.

In the Transformer architecture, the representation of the source sequence is supplied to the decoder through the encoder-decoder attention. This attention sub-layer is applied between the self-attention and feed-forward sub-layers in each Transformer layer. Such arrangement leaves many options for the incorporation of multiple encoders.

So far, attention in sequence-to-sequence learning with multiple source sequences was mostly studied in the context of recurrent neural networks (RNNs). Libovický and Helcl (2017) explicitly capture the distribution over multiple inputs by projecting the input representations to a shared vector space and either computing the attention over all hidden states at once, or hierarchically, using another level of attention applied on the context vectors. Zoph and Knight (2016) employ a gating mechanism for combining the context vectors. Voita et al. (2018) adapted the gating mechanism for use within the Transformer model for context-aware MT. The other aproaches are however not directly usable in the Transformer model.

We propose a number of strategies of combining the different sources in the Transformer model. Some of the strategies described in this work are an adaptation of the strategies previously used with recurrent neural networks (Libovický and Helcl, 2017), whereas the rest of them is a novel contribution devised for the Transformer architecture. We test these strategies on multimodal machine translation (MMT) and multi-source machine translation (MSMT) tasks.

This paper is organized as follows. In Section 2, we briefly describe the decoder part of the Transformer model. We propose a number of input combination strategies for the multi-source Transformer model in Section 3. Section 4 describes the experiments we performed, and Section 5 shows the results of quantitative evaluation. An overview of the related work is given in Section 6. We discuss the results and conclude in Section 7.

## 2 Transformer Decoder

The Transformer architecture is based on the use of attention. Attention, as conceptualized by Vaswani et al. (2017), can be viewed as a soft-lookup function operating on an associative memory. For each query vector in query set $Q$, the attention computes a set of weighted sums of values $V$ associated with a set of keys $K$, based on their similarity to the query.

The variant of the attention function used in the Transformer architecture is called *multi-head scaled dot-product* attention. Scaled dot-product

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 253–260
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64026

of queries and keys is used as the similarity measure. Given the dimension of the input vectors $d$, the attention is computed as follows:

$$\mathcal{A}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V. \quad (1)$$

In the multi-head variant, the vectors that represent the queries, keys, and values are linearly transformed to a number of projections (usually with smaller dimension), called *attention heads*. The attention is computed in each head independently and the outputs are concatenated and projected back to the original dimension:

$$\mathcal{A}^h(Q, K, V) = \sum_{i=1}^{h} C_i W_i^O \quad (2)$$

where $W_i^O \in \mathbb{R}^{d_h \times d}$ are trainable parameter matrices used as projections of the attention head outputs of dimension $d_h$ to the model dimension $d$, and

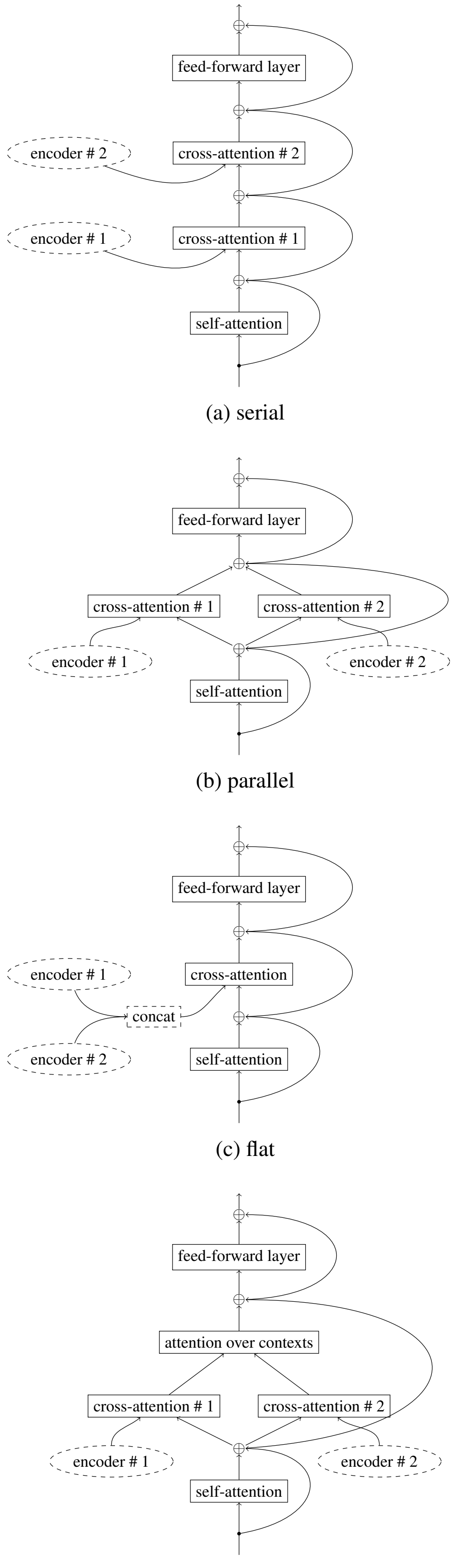
$$C_i = \mathcal{A}(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

where $W^Q$, $W^K$, and $W^V \in \mathbb{R}^{d \times d_h}$, are trainable projection matrices used to project the attention inputs to the attention heads.

The model itself consists of a number of layers, each of which is divided in three sub-layers: self-attention, encoder-decoder (or cross) attention, and a feed-forward layer. Both of the attention types use identical sets for keys and values. The states of the previous layer are used as the query set. The self-attention sub-layer attends to the previous decoder layer (i.e. the sets of queries and keys are identical). Since the decoder works autoregressively from left to right, during training, the self-attention is masked to prevent attending to the future positions in the sequence. The encoder-decoder attention sub-layer attends to the final layer of the encoder. The feed-forward sub-layer consists of a single non-linear projection (usually to a space with larger dimension), followed by a linear projection back to the vector space with the original dimension. The input of each sub-layer is summed with the output, creating a residual connection chain throughout the whole layer stack.

## 3 Proposed Strategies

We propose four input combination strategies for multi-source variant of the Transformer network,



(a) serial

(b) parallel

(c) flat

(d) hierarchical

Figure 1: Schemes of computational steps for the serial, parallel, flat, and hierarchical attention combination in a single layer of the decoder.

as illustrated in Figure 1. Two of them, serial and parallel, model the encoder-decoder attentions independently and are a natural extension of the sub-layer scheme in the transformer decoder. The other two versions, flat and hierarchical, are inspired by approaches proposed for RNNs by Libovický and Helcl (2017) and model joint distributions over the inputs.

**Serial.** The serial strategy (Figure 1a) computes the encoder-decoder attention one by one for each input encoder. The query set of the first cross-attention is the set of the context vectors computed by the preceding self-attention. The query set of each subsequent cross-attention is the output of the preceding sub-layer. All of these sub-layers are interconnected with residual connections.

**Parallel.** In the parallel combination strategy (Figure 1b), the model attends to each encoder independently and then sums up the context vectors. Each encoder is attended using the same set of queries, i.e. the output of the self-attention sub-layer. Residual connection link is used between the queries and the summed context vectors from the parallel attention.

$$\mathcal{A}^h_{para}(Q, K_{1:n}, V_{1:n}) = \sum_{i=1}^{n} \mathcal{A}^h(Q, K_i, V_i) \quad (4)$$

**Flat.** The encoder-decoder attention in the flat combination strategy (Figure 1c) uses all the states of all input encoders as a single set of keys and values. Thus, the attention models a joint distribution over a flattened set of all encoder states. Unlike the approach taken in the recurrent setup (Libovický and Helcl, 2017), where the flat combination strategy requires an explicit projection of the encoder states to a shared vector space, in the Transformer models, the vector spaces of all layers are tied with residual connections. Therefore, the intermediate projection of the states of each encoder is not necessary.

$$K_{flat} = V_{flat} = \mathrm{concat}_i(K_i) \quad (5)$$

$$\mathcal{A}^h_{flat}(Q, K_{1:n}, V_{1:n}) = \mathcal{A}^h(Q, K_{flat}, V_{flat}) \quad (6)$$

**Hierarchical.** In the hierarchical combination (Figure 1d), we first compute the attention independently over each input. The resulting contexts are then treated as states of another input and the attention is computed once again over these states.

$$K_{hier} = V_{hier} = \mathrm{concat}_i(\mathcal{A}^h(Q, K_i, V_i)) \quad (7)$$

$$\mathcal{A}^h_{hier}(Q, K_{1:n}, V_{1:n}) = \mathcal{A}^h(Q, K_{hier}, V_{hier}) \quad (8)$$

## 4 Experiments

We conduct our experiments on two different tasks: multimodal translation and multi-source machine translation. We use Neural Monkey (Helcl and Libovický, 2017)[1] for design, training, and evaluation of the experiments.

In all experiments, the encoder part of the network follows the Transformer architecture as described by Vaswani et al. (2017).

We optimize the model parameters using Adam optimizer (Kingma and Ba, 2014) with initial learning rate 0.2, and Noam learning rate decay (Vaswani et al., 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$, and 4,000 warm-up steps. The size of a mini-batch size of 32 for MMT, and 24 for multi-source MT experiments.

During decoding, we use beam search of width 10 and length normalization of 1.0 (Wu et al., 2016).

### 4.1 Multimodal Translation

The goal of MMT (Specia et al., 2016) is translating image captions from one language into another given both the source and image as the input. We use Multi30k dataset (Elliott et al., 2016) containing triplets of images, English captions and their English translations into German, French and Czech. The dataset contains 29k triplets for training, 1,014 for validation and a test set of 1,000. We experiment with all language pairs available in this dataset.

We extract image feature using the last convolutional layer of the ResNet network (He et al., 2016) trained for ImageNet classification. We apply a linear projection into 512 dimensions on the image representation, so it has the same dimension as the rest of the model. For each language pair, we create a shared wordpiece-based vocabulary of approximately 40k subwords. We share the embedding matrices across the languages and we use the transposed embedding matrix as the output projection matrix as proposed by Press and Wolf (2017).

We use 6 layers in the textual encoder and decoder, and set the model dimension to 512. We

[1] `http://github.com/ufal/neuralmonkey`

set the dimension of the hidden layers in the feed-forward sub-layers to 4096. We use 16 heads in the attention layers.

During the evaluation, we follow the preprocessing used in WMT Multimodal Translation Shared Task (Specia et al., 2016).

Conclusions of previous work show (Elliott and Kádár, 2017) that the improved performance of the multimodal models compared to textual models can come from improving the input representation. In order to test whether it is also the case with our models or the models explicitly use the visual input, we perform an adversarial evaluation similar to Elliott (2018). We evaluate the model while providinng a random image and observe how it affects the score and observe whether their quality drops.

### 4.2 Multi-Source MT

In this set of experiment, we attempt to generate a sentence in a target language, given equivalent sentences in multiple source languages.

We use the Europarl corpus (Tiedemann, 2012) for training and testing the MSMT. We use Spanish, French, German, and English as source languages and Czech as a target language. We selected an intersection of the bilingual sub-corpora using English as a pivot language. Our dataset contains 511k 5-tuples of sentences for training, 1k for validation and another 1k for testing.

Due of the memory demands of having four encoders, we use a smaller model than in the previous experiment. The encoders only have 4 layers and the decoder has 6 layers with embeddings size 256, feed-forward layers dimension 2048, and 8 attention heads. We use a shared word-piece vocabulary of 48k subwords. As in the MMT experiments, the transposition of the embedding matrix is reused as the parameters of the output projection layer (Press and Wolf, 2017).

We use bilingual English-to-Czech translation as a single source baseline. The baseline uses vocabulary of 42k subwords from Czech and English only.

Similarly to the MMT, we also perform adversarial evaluation. To evaluate the importance of the source languages for the translation quality, when randomizing one of the source languages.

## 5 Results

We evaluate the results using BLEU (Papineni et al., 2002) and METEOR (Denkowski and Lavie, 2011) as implemented in MultEval. [2] The results of the MMT task are tabulated in Table 1. The results of the multi-source MT are shown in Table 2.

In MMT, the input combination significantly surpassed the text-only baseline in English-to-French translation. The performance in other target languages is only slightly better than the textual baseline.

The only worse score was achieved by the flat combination strategy. We hypothesize this might be because the optimization failed to find a common representation of the input modalities that could be used to compute the joint distribution.

The adversarial evaluation with randomly selected input images shows that all our models rely on both inputs while generating the target sentence and that providing incorrect visual input harms the model performance. The modality gating in the hierarchical attention combination seems to make the models more robust to noisy visual input.

In the multi-source translation task, all the proposed strategies perform better than single-source translation from English to Czech. Among the combination strategies, the best-scoring is the serial stacking of the attentions. In multimodal translation, the flat combination has shown to be the best-performing strategy.

Analysis of the attention distribution shows that the serial strategy use information from all source languages. The parallel strategy almost does not use the Spanish source and the flat strategy prefers the English source. The hierarchical strategy uses information from all source languages, however the attentions are sometimes more fuzzy than in the previous strategies. Figure 2 shows what source languages were attended on different layers of the encoder. Other examples of the attention visualization are shown in Appendix A.

The adversarial evaluation shows all the models used English as a primary source. Providing incorrect English source harms. Introducing noise into other languages affects the score in much smaller scale.

|  | MMT: en→de |  |  | MMT: en→fr |  |  | MMT: en→cs |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | BLEU | METEOR | adv.BLEU | BLEU | METEOR | adv.BLEU | BLEU | METEOR | adv.BLEU |
| baseline | 38.3 ± .8 | 56.7 ± .7 | — | 59.6 ± .9 | 72.7 ± .7 | — | 30.9 ± .8 | 29.5 ± .4 | — |
| serial | 38.7 ± .9 | 57.2 ± .6 | 37.3 ± .6 | 60.8 ± .9 | 75.1 ± .6 | 58.9 ± .9 | 31.0 ± .8 | 29.9 ± .4 | 29.7 ± .8 |
| parallel | 38.6 ± .9 | 57.4 ± .7 | 38.2 ± .8 | 60.2 ± .9 | 74.9 ± .6 | 58.9 ± .9 | 31.1 ± .9 | 30.0 ± .4 | 30.4 ± .8 |
| flat | 37.1 ± .8 | 56.5 ± .6 | 35.7 ± .8 | 58.0 ± .9 | 73.3 ± .7 | 57.0 ± .9 | 29.9 ± .8 | 29.0 ± .4 | 28.2 ± .8 |
| hierarchical | 38.5 ± .8 | 56.5 ± .6 | 38.1 ± .8 | 60.8 ± .9 | 75.1 ± .6 | 60.2 ± .9 | 31.3 ± .9 | 30.0 ± .4 | 31.0 ± .8 |

Table 1: Quantitative results of the MMT experiments on the 2016 test set. Column 'adv. BLEU' is an adversarial evaluation with randomized image input.

|  | MSMT |  | Adversarial evaluation (BLEU) |  |  |  |
|---|---|---|---|---|---|---|
|  | BLEU | METEOR | en | de | fr | es |
| baseline | 16.5 ± .5 | 20.5 ± .3 | — | — | — | — |
| serial | 20.5 ± .6 | 23.5 ± .5 | 8.1 ± .4 | 19.7 ± .5 | 19.5 ± .6 | 18.4 ± .5 |
| parallel | 20.5 ± .6 | 23.3 ± .3 | 1.4 ± .2 | 18.7 ± .5 | 17.9 ± .5 | 20.3 ± .5 |
| flat | 20.4 ± .6 | 23.3 ± .3 | 0.2 ± .1 | 19.9 ± .6 | 20.0 ± .6 | 19.6 ± .5 |
| hierarchical | 19.4 ± .5 | 22.7 ± .3 | 4.2 ± .3 | 18.3 ± .5 | 18.3 ± .5 | 15.3 ± .5 |

Table 2: Quantitative results of the MMT experiment. The adversarial evaluation shows the BLEU score when one input language was changed randomly.

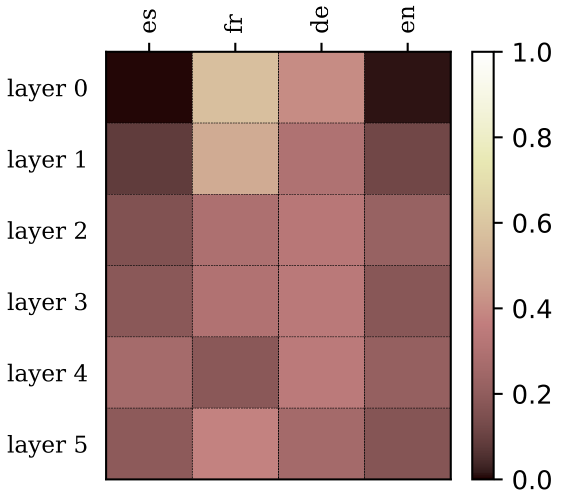


Figure 2: Attention over contexts in the hiearchical strategy over the decoder layers.

## 6 Related Work

MMT was so far solved only within the RNN-based architectures. Elliott et al. (2015) report significant improvements with a non-attentive model. With attentive models (Bahdanau et al., 2014), the additional visual information usually did not improve the models significantly (Caglayan et al., 2016; Helcl and Libovický, 2017) in terms of BLEU score. Our models slightly outperform these models in the single model setup.

Except for using the image features direct input to the model, they can be used as an auxiliary objective (Elliott and Kádár, 2017). In this setup, the visually grounded representation, improves the MMT significantly, achieving similar results that our models achieved using only the Multi30k dataset.

To our knowledge, multi-source MT has also been studied only using the RNN-based models. Dabre et al. (2017) use simple concatenation of source sentences in various languages and process them with a single multilingual encoder.

Zoph and Knight (2016) try context concatenation and hierarchical gating method for combining context vectors in attention models with multiple inputs encoded by separate encoders. In all of their experiments, the multi-source methods significantly surpass the single-source baseline. Nishimura et al. (2018) extend the former approach for situations when of the source languages is missing, so that the translation system does not overly rely on a single source language like some of the models presented in this work.

## 7 Conclusions

We proposed several input combination strategies for multi-source sequence-to-sequence learning using the Transformer model (Vaswani et al., 2017). Two of the strategies are a straightforward extension of cross-attention in the Trans-

[2] https://github.com/jhclark/multeval

former model: the cross-attentions are combined either serially interleaved by residual connections or in parallel. The two remaining strategies are an adaptation of the flat and the hierarchical attention combination strategies introduced by Libovický and Helcl (2017) in context of recurrent sequence-to-sequence models.

The results on the MMT task show similar properties an in RNN-based models (Caglayan et al., 2017; Libovický and Helcl, 2017). Adding visual features significantly improves translation into French and brings minor improvements on other language pairs. All the attention combinations perform similarly with the exception of the flat strategy which probably struggles with learning a shared representation of the input tokens and the image representation.

Evaluation on multi-source MT shows significant improvements over the single-source baseline. However, the adversarial evaluation suggests that the model relies heavily on the English input and only uses the additional source languages for minor modifications of the output. All attention combinations performed similarly.

## Acknowledgments

This research received support from the grant No. 18-02196S and No. P103/12/G084 of the Grant Agency of the Czech Republic, and the grant No. 976518 of the Grant Agency of the Charles University. This research was partially supported by SVV project number 260 453.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost van de Weijer. 2017. Lium-cvc submissions for wmt17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 432–439, Copenhagen, Denmark. Association for Computational Linguistics.

Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost van de Weijer. 2016. Does multimodality help human and machine for translation and image captioning? In *Proceedings of the First Conference on Machine Translation*, pages 627–633, Berlin, Germany. Association for Computational Linguistics.

Raj Dabre, Fabien Cromierès, and Sadao Kurohashi. 2017. Enabling multi-source neural machine translation by concatenating source sentences in multiple languages. *CoRR*, abs/1702.06135.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, United Kingdom. Association for Computational Linguistics.

Desmond Elliott. 2018. Adversarial evaluation of multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, and Eva Hasler. 2015. Multi-language image description with neural sequence models. *CoRR*, abs/1510.04709.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Desmond Elliott and Ákos Kádár. 2017. Imagination improves multimodal translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 130–141, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. IEEE Computer Society.

Jindřich Helcl and Jindřich Libovický. 2017. Neural Monkey: An open-source tool for sequence learning. *The Prague Bulletin of Mathematical Linguistics*, 107:5–17.

Jindřich Helcl and Jindřich Libovický. 2017. CUNI system for the WMT17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 450–457, Copenhagen, Denmark. Association for Computational Linguistics.

Lukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017. One model to learn them all. *CoRR*, abs/1706.05137.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202, Vancouver, Canada. Association for Computational Linguistics.

Yuta Nishimura, Katsuhito Sudoh, Graham Neubig, and Satoshi Nakamura. 2018. Multi-source neural machine translation with missing data. In *The Second Workshop on Neural Machine Translation and Generation (WNMT)*, Melbourne, Australia.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163. Association for Computational Linguistics.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation*, pages 543–553, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

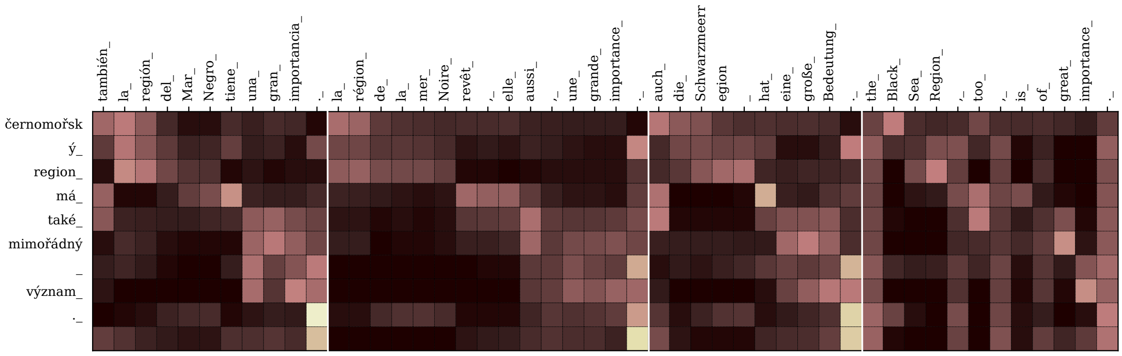
Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

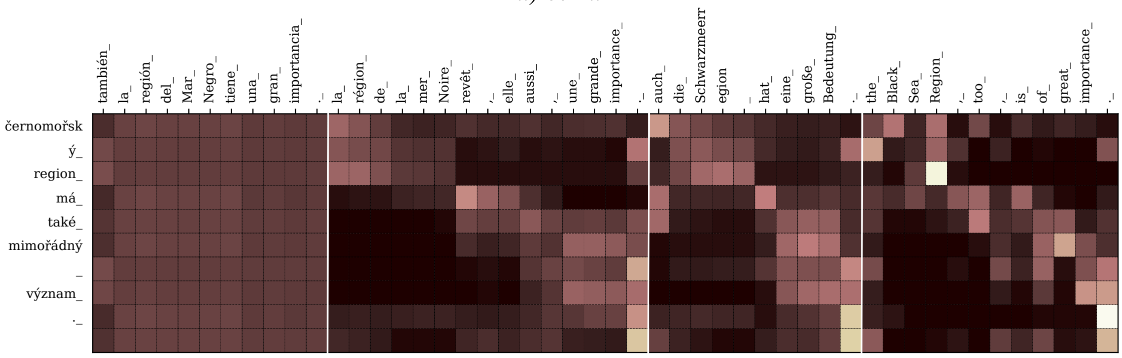
Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.

## A Attention Visualizations

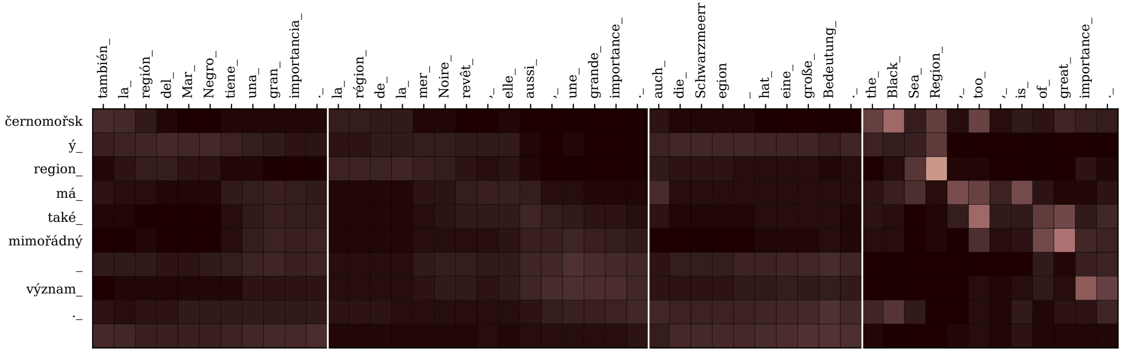
We show cross-attention visualizations for the four proposed combination strategies on Multi-source MT. The Czech target wordpieces are in rows, the source Spanish, French, German, and English wordpieces are concatenated and shown in columns. These attentions were taken form the decoder's fourth layer and were averaged across the individual heads. For serial and parallel strategy the cross-attention weights sum to one for each language separately, the flat strategy has only one common cross-attention, and for the hierarchical strategy visualization the cross-attention weights for individual languages are multiplied by the weights of the attention over contexts.



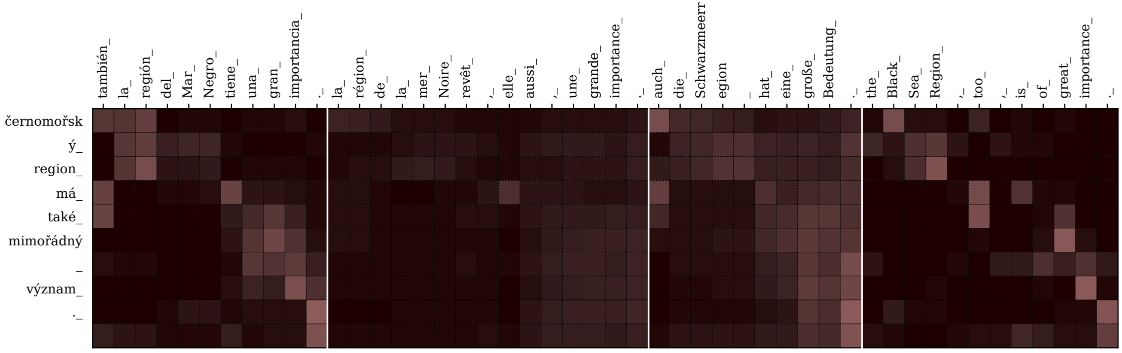
a) serial



b) parallel



c) flat



d) hierarchical

# Parameter Sharing Methods for Multilingual Self-Attentional Translation Models

**Devendra Singh Sachan**
Data Solutions Team
Petuum Inc.
Pittsburgh, USA
devendra.singh@petuum.com

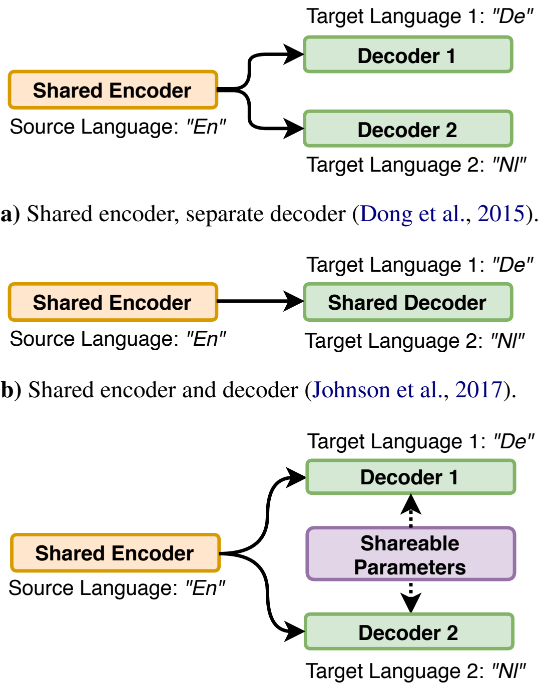
**Graham Neubig**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, USA
gneubig@cs.cmu.edu

## Abstract

In multilingual neural machine translation, it has been shown that sharing a single translation model between multiple languages can achieve competitive performance, sometimes even leading to performance gains over bilingually trained models. However, these improvements are not uniform; often multilingual parameter sharing results in a decrease in accuracy due to translation models not being able to accommodate different languages in their limited parameter space. In this work, we examine parameter sharing techniques that strike a happy medium between full sharing and individual training, specifically focusing on the self-attentional *Transformer* model. We find that the full parameter sharing approach leads to increases in BLEU scores mainly when the target languages are from a similar language family. However, even in the case where target languages are from different families where full parameter sharing leads to a noticeable drop in BLEU scores, our proposed methods for partial sharing of parameters can lead to substantial improvements in translation accuracy.[1]

## 1 Introduction

Neural machine translation (NMT; Sutskever et al. (2014); Cho et al. (2014)) is now the de-facto standard in MT research due to its relative simplicity of implementation, ability to perform end-to-end training, and high translation accuracy. Early approaches to NMT used recurrent neural networks (RNNs), usually LSTMs (Hochreiter and Schmidhuber, 1997), in their encoder and decoder layers, with the addition of an attention mechanism (Bahdanau et al., 2014; Luong et al., 2015) to focus more on specific encoded source words when deciding the next translation target output. Recently, the NMT research community has been transitioning from RNNs to an alternative method for encoding sentences using self-attention (Vaswani et al., 2017), represented by the so-called "*Transformer*" model, which both improves the speed of processing sentences on computational hardware such as GPUs due to its lack of recurrence, and achieves impressive results.



**(a)** Shared encoder, separate decoder (Dong et al., 2015).

**(b)** Shared encoder and decoder (Johnson et al., 2017).

**(c)** Proposed shared decoder with partial parameter sharing.

**Figure 1:** Examples of MTL frameworks for the translation of one source language (for example *"En"*) to two target languages (for example *"De"*, *"Nl"*). The principle remains the same with more than two target languages. Best viewed in color.

In parallel to this transition to self-attentional models, there has also been an active interest in the multilingual training of NMT systems (Firat et al., 2016; Johnson et al., 2017; Ha et al.,

[1] Data and code of this paper is available at: https://github.com/DevSinghSachan/multilingual_nmt

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 1: Research Papers, pages 261–271
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64027

2016). In contrast to the standard bilingual models, multilingual models follow the multi-task training paradigm (Caruana, 1997) where models are *jointly trained* on training data from several language pairs, with some degree of *parameter sharing*. The objective of this is two-fold: First, compared to individually training separate models for each language pair of interest, this maintains competitive translation accuracy while reducing the total number of models that need to be stored, a considerable advantage when deploying practical systems. Second, by utilizing data from multiple language pairs simultaneously, it becomes possible to improve the translation accuracy for each language pair.

In multilingual translation, *one-to-many* translation —translation from a common source language (for example English) to multiple target languages (for example German and Dutch) — is considered particularly difficult. Previous multi-task learning (MTL) models for this task broadly consist of two approaches as shown in Figure 1: (a) a model with a shared encoder and one decoder per target language (Dong et al. (2015), shown in Figure 1a). This approach has the advantage of being able to model each target separately but comes with the cost of slower training and increased memory requirements. (b) a single *unified* model consisting of a shared encoder and a shared decoder for all the language pairs (Johnson et al. (2017), shown in Figure 1b). This simple approach is trivially implementable using a standard bilingual translation model and has the advantage of having a constant number of trainable parameters regardless of the number of languages, but has the caveat that the decoder's ability to model multiple languages can be significantly reduced.

In this paper, we propose a third alternative: (c) a model with a shared encoder and multiple decoders such that some decoder parameters are shared (shown in Figure 1c). This hybrid approach combines the advantages from both the approaches mentioned above. It carefully moderates the types of parameters that are shared between the multiple languages to provide the flexibility necessary to decode two different languages, but still shares as many parameters as possible to take advantage of information sharing across multiple languages. Specifically, we focus on the aforementioned self-attentional Transformer models, with the set of shareable parameters consisting of the various attention weights, linear layer weights, or embedding weights contained therein. The *full sharing* and *no sharing* of decoder parameters used in previous work are special cases (refer to Section 2.2 for a detailed description).

To empirically examine the utility of this approach, we examine the case of translation from a common source language to multiple target languages, where the target languages can be either related or unrelated. Our work reveals that while full parameter sharing works reasonably well when using target languages from the same family, partial parameter sharing is essential to achieve the best accuracy when translating into multiple distant languages.

## 2 Method

In this section, we will first briefly describe the key elements of the Transformer model followed by our proposed approach of parameter sharing.

### 2.1 Transformer Architecture

As is common in sequence-to-sequence (*seq2seq*) models for NMT, the self-attentional Transformer model (Figure 2; Vaswani et al. (2017)) consists of an embedding layer, multiple encoder-decoder layers, and an output generation layer. Each encoder layer consists of two sublayers in sequence: self-attentional and feed-forward networks. Each decoder layer consists of three sublayers: masked self-attention, encoder-decoder attention, and feed-forward networks. The core building blocks in all these layers consist of different sets of weight matrices that compute affine transforms.

First, an embedding layer obtains the source and target word vectors from the input words: $\boldsymbol{W}_E \in \mathbb{R}^{d_m \times V}$, where $d_m$ is model size, and $V$ is vocabulary size. After the embedding lookup step, word vectors are multiplied by a scaling factor of $\sqrt{d_m}$. To capture the relative position of a word in the input sequence, *position encodings* defined in terms of sinusoids of different frequencies are added to the scaled word vectors of the source and target.

The encoder layer maps the input word vectors to continuous hidden state representations. As mentioned earlier, it consists of two sublayers. The first sublayer performs *multi-head dot-product self-attention*. In the single-head case, defining the input to the sublayer as $x = (x_1, \ldots, x_T)$ and the output as $z = (z_1, \ldots, z_T)$, where $x_i, z_i \in \mathbb{R}^{d_m}$,

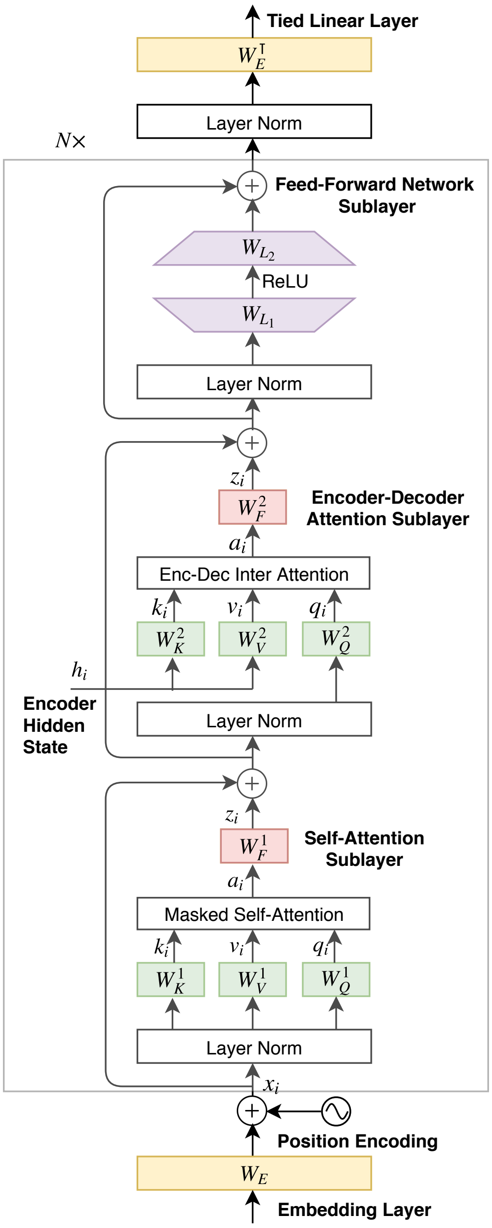


**Figure 2:** Block diagram illustrating the Transformer decoder's shareable parameters (in color) that includes embedding layer weights ($\boldsymbol{W_E}$), tied linear layer weights ($\boldsymbol{W_E^T}$), transformation weights as a part of self-attention ($\boldsymbol{W_K^1}, \boldsymbol{W_V^1}, \boldsymbol{W_Q^1}, \boldsymbol{W_F^1}$), encoder-decoder attention ($\boldsymbol{W_K^2}, \boldsymbol{W_V^2}, \boldsymbol{W_Q^2}, \boldsymbol{W_F^2}$), and feed-forward network ($\boldsymbol{W_{L_1}}, \boldsymbol{W_{L_2}}$) sublayers. Best viewed in color.

the input is linearly transformed to obtain key ($k_i$), value ($v_i$), and query ($q_i$) vectors

$$k_i = x_i \boldsymbol{W}_K, v_i = x_i \boldsymbol{W}_V, q_i = x_i \boldsymbol{W}_Q.$$

Next, similarity scores ($e_{ij}$) between query and key vectors are computed by performing a scaled dot-product

$$e_{ij} = \frac{1}{\sqrt{d_m}} q_i k_j^T.$$

Next, attention coefficients ($\alpha_{ij}$) are computed by applying softmax function over these similarity values.

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{l=1}^{T} \exp e_{il}}$$

Self-attention output ($z_i$) is computed by the convex combination of attention weights with value vectors followed by a linear transformation

$$z_i = (\sum_{j=1}^{T} \alpha_{ij} v_j) \boldsymbol{W}_F.$$

In the above equations, $\boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{W}_Q, \boldsymbol{W}_F$ are learnable transformation matrices of shape $\mathbb{R}^{d_m \times d_m}$. To extend to multi-head attention ($\ell$), one can split the key, value, and query vectors into $\ell$ vectors, perform the attention computation in parallel for each of the $\ell$ vectors followed by concatenating before the final linear transformation by $\boldsymbol{W}_F$. The second sublayer consists of a two-layer deep *position-wise feed-forward network* (FFN) with ReLU activation (Glorot et al., 2011).

$$\text{FFN}(z_i) = \max(0,\ z_i \boldsymbol{W_{L_1}} + b_1) \boldsymbol{W_{L_2}} + b_2$$

where $\boldsymbol{W_{L_1}} \in \mathbb{R}^{d_m \times d_h}$, $\boldsymbol{W_{L_2}} \in \mathbb{R}^{d_h \times d_m}$, $b_1$ and $b_2$ are biases, and $d_h$ is hidden size. The FFN sublayer outputs are subsequently given as input to the next encoder layer.

The decoder layer consists of three sublayers. The first sublayer, similar to the encoder, performs masked self-attention where masks are used to prevent positions from attending to subsequent positions. The second sublayer performs *encoder-decoder inter-attention* where the input to the query vector comes from the decoder layer while the input to the key and value vectors comes from the encoder's last layer. To denote parameters in these two sublayers, the transformation weights of the masked self-attention sublayer are referenced as $\boldsymbol{W_K^1}, \boldsymbol{W_V^1}, \boldsymbol{W_Q^1}, \boldsymbol{W_F^1}$ and encoder-decoder attention sublayer as $\boldsymbol{W_K^2}, \boldsymbol{W_V^2}, \boldsymbol{W_Q^2}, \boldsymbol{W_F^2}$, which is also indicated in Figure 2. The third sublayer consists of an FFN. To generate predictions for the next word, there is a linear layer on top of the decoder layer. The weight of this linear layer is shared with the weight of the embedding layer (Inan et al., 2016).

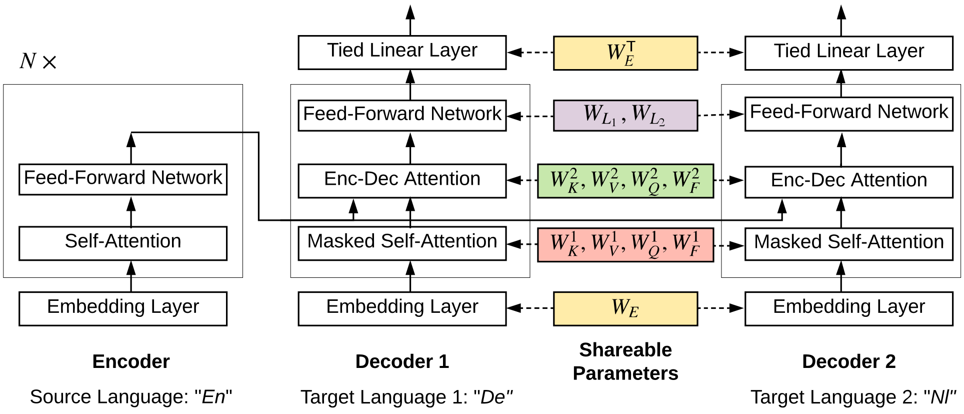


**Figure 3:** Block diagram illustrating our MTL approach for *one-to-many* multilingual translation task that is based on the partial sharing of parameters between the multiple decoders. Best viewed in color.

Residual connections (He et al., 2016) and layer normalization (Ba et al., 2016) are applied on each sublayer and to the output vector from the final encoder and decoder layers.

## 2.2 Parameter Sharing Strategies

In this paper, our objective is to investigate effective parameter sharing strategies for the Transformer model using MTL, mainly for *one-to-many* multilingual translation. Here, we will use the symbol $\boldsymbol{\Theta}$ to denote the set of shared parameters in our model. These parameter sharing strategies are described below:

- The base case consists of separate bilingual translation models for each language pair $\left(\boldsymbol{\Theta} = \emptyset\right)$.

- Use of a common embedding layer for all the bilingual models $\left(\boldsymbol{\Theta} = \{\boldsymbol{W}_E\}\right)$. This will result in a significant reduction of the total parameters by sharing parameters across common words present in the source and target sentences (Wu et al., 2016).

- Use of a common encoder for the source language and a separate decoder for each target language $\left(\boldsymbol{\Theta} = \{\boldsymbol{W}_E, \boldsymbol{\theta}_{ENC}\}\right)$. This has the advantage that the encoder will now see more source language training data (Dong et al., 2015).

Next, we also include the decoder parameters among the set of shared parameters. While doing so, we will assume that the embedding and the encoder parameters are always shared between the bilingual models. Because there can be exponentially many combinations considering all the different feasible sets of shared parameters between the multiple decoders, we only select a subset of these combinations based on our preliminary results. These selected weights are shared in all the layers of the decoder unless stated otherwise. A schematic diagram illustrating the various possible parameter matrices that can be shared in each sublayer of our MTL model is shown in Figure 3.

- We share only the FFN sublayer parameters $\left(\boldsymbol{\Theta} = \left\{\boldsymbol{W}_E, \boldsymbol{\theta}_{ENC}, \boldsymbol{W}_{L_1}, \boldsymbol{W}_{L_2}\right\}\right)$.

- Sharing the weights of the self-attention sublayer $\left(\boldsymbol{\Theta} = \left\{\boldsymbol{W}_E, \boldsymbol{\theta}_{ENC}, \boldsymbol{W}_K^1, \boldsymbol{W}_Q^1, \boldsymbol{W}_V^1, \boldsymbol{W}_F^1\right\}\right)$.

- Sharing the weights of the encoder-decoder attention sublayer $\left(\boldsymbol{\Theta} = \left\{\boldsymbol{W}_E, \boldsymbol{\theta}_{ENC}, \boldsymbol{W}_K^2, \boldsymbol{W}_Q^2, \boldsymbol{W}_V^2, \boldsymbol{W}_F^2\right\}\right)$.

- We limit the attention parameters that are shared to only include either the key and query weights $\left(\boldsymbol{\Theta} = \left\{\boldsymbol{W}_E, \boldsymbol{\theta}_{ENC}, \boldsymbol{W}_K^1, \boldsymbol{W}_Q^1, \boldsymbol{W}_K^2, \boldsymbol{W}_Q^2\right\}\right)$ or the key and value weights $\left(\boldsymbol{\Theta} = \left\{\boldsymbol{W}_E, \boldsymbol{\theta}_{ENC}, \boldsymbol{W}_K^1, \boldsymbol{W}_V^1, \boldsymbol{W}_K^2, \boldsymbol{W}_V^2\right\}\right)$. The motivation for doing so is so that the shared attention sublayer weights can model the common aspects of the target languages while the individual FFN sublayer weights can model the distinctive or unique aspects of each language.

- We share all the parameters of the decoder to have a single unified model $\left(\boldsymbol{\Theta} = \left\{\boldsymbol{W}_E, \boldsymbol{\theta}_{ENC},\right.\right.$

| Language Pair | Training | Dev | Test |
|---|---|---|---|
| En–Ro | 180,484 | 3,904 | 4,631 |
| En–Fr | 192,304 | 4,320 | 4,866 |
| En–Nl | 183,767 | 4,459 | 5,006 |
| En–De | 167,888 | 4,148 | 4,491 |
| En–Ja | 204,090 | 4,429 | 5,565 |
| En–Tr | 182,470 | 4,045 | 5,029 |

**Table 1:** Number of sentences in the training, dev, and test splits for each language pair used in our experiments. The languages are represented by their ISO 639-1 codes *En:English*, *Fr:French*, *Nl:Dutch*, *De:German*, *Ja:Japanese*, *Tr:Turkish*.

$\boldsymbol{\theta}_{DEC}\})$. Fewer parameters in the decoder indicates limited modeling ability, and we expect this method to obtain good translation accuracy mainly when the target languages are related (Johnson et al., 2017).

## 3 Experimental Setup

In this section, first, we describe the datasets used in this work and the evaluation criteria. Then, we describe the training regimen followed in all our experiments. All of our models were implemented in PyTorch framework (Paszke et al., 2017) and were trained on a single GPU.

### 3.1 Datasets and Evaluation Metric

To perform multilingual translation experiments, we select six language pairs from the openly available TED talks dataset (Qi et al., 2018) whose statistics are mentioned in Table 1. This dataset already contains predefined splits for training, development, and test sets. Among these languages, Romanian (Ro) and French (Fr) are *Romance* languages, German (De) and Dutch (Nl) are *Germanic* languages while Turkish (Tr) and Japanese (Ja) are unrelated languages that come from distant language families. For all language pairs, tokenization was carried out using the `Moses` tokenizer,[2] except for Japanese, where word segmentation was performed using the `KyTea` tokenizer (Neubig et al., 2011). To select training examples, we filter sentences with a maximum length of 70 tokens. For evaluation, we report the model's performance using the standard BLEU score metric (Papineni et al., 2002). We use the `mtevalv14.pl` script from the `Moses` toolkit to compute the tokenized BLEU scores.

### 3.2 Training Protocols

In this work, we follow the same training process for all the experiments. We jointly encode the source and target language words with subword units by applying *byte pair encoding* (Gage, 1994) with 32,000 merge operations (Sennrich et al., 2016). These subword units restrict the vocabulary size and prevent the need for explicitly handling out-of-vocabulary symbols as the vocabulary can be used to represent any word. We use *LeCun uniform initialization* (LeCun et al., 1998) for all the trainable model parameters. Embedding layer weights are randomly initialized according to truncated Gaussian distribution $\boldsymbol{W}_E \sim \mathcal{N}(0, d_m{}^{-1/2})$.

In all the experiments, we use *Transformer base model* configuration (Vaswani et al., 2017) that consists of six encoder-decoder layers, $d_m = 512$, $d_h = 2,048$, and $\ell = 8$. For optimization, we use SGD with Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.997$, and $\epsilon = 1e^{-9}$.[3] The learning rate (*lr*) schedule is varied at every optimization step (*step*) according to:

$$lr = 2d_m^{-0.5}\text{min}\left(step^{-0.5}, step \cdot 16000^{-1.5}\right)$$

Each mini-batch consists of approximately $3,000$ source and $3,000$ target tokens such that similar length sentences are bucketed together. We train the models until convergence and save the best checkpoint using development set performance. For model regularization, we use label smoothing $(\epsilon = 0.1)$ (Pereyra et al., 2017) and apply dropout (with $p_{drop} = 0.1$) (Srivastava et al., 2014) to the word embeddings, attention coefficients, ReLU activation, and to the output of each sublayer before the residual connection. During decoding, we use beam search with beam width 5 and length normalization with $\alpha = 1$ (Wu et al., 2016).

### 3.3 Multilingual Training

During the multilingual model's training and inference, we include an additional token representing the desired target language at the start of each source sentence (Johnson et al., 2017). The presence of this additional token will help the model learn the target language to translate to during decoding. For preprocessing, we apply byte pair en-

[2] https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer

[3] These hyperparameter values are based on the single GPU transformer model from the open-source *tensor2tensor* toolkit.

coding over the combined dataset of all the language pairs. We perform model training using balanced mini-batches *i.e.* it contains roughly an equal number of sentences for every target language. While training, we compute weighted average cross-entropy loss where the weighting term is proportional to the total word count observed in each of the target language sentences.

## 4 Results

In this section, we will describe the results of our proposed parameter sharing techniques and later present the broader context by comparing them with bilingual translation models and previous benchmark methods.

### 4.1 Parameter Sharing

Here, we first analyze the results of *one-to-many* multilingual translation experiments when there are two target languages and both of them belong to the same language family. The first set of experiments are on *Romance* languages (En→Ro+Fr) and the second set of experiments are on *Germanic* languages (En→De+Nl). We report the BLEU scores in Table 2a when different sets of parameters are shared in these experiments. We observe that sharing only the embedding layer weight between the multiple models leads to the lowest scores. Sharing the encoder weights results in significant improvement for En→Ro+Fr but leads to a small decrease in En→De+Nl scores.

We then gradually include both the decoder's weights to the set of shareable parameters. Specifically, we include the parameters of FFN, self-attention, encoder-decoder attention, both the attention sublayers, key, query, value weights from both the attention sublayers, and finally all the parameters of the decoder layer. From the results, we note that the sharing of the encoder-decoder attention weights leads to substantial gains. Finally, sharing the entirety of the parameters (*i.e.* having one model) leads to the best BLEU scores for En→Ro+Fr and sharing only the key and query matrices from both the attention layers leads to the best BLEU scores for En→De+Nl. One of the reasons for such large increase in BLEU is that encoder has access to more English language training data and for the decoder, as the target languages belong to the same family, they may contain common vocabulary, thus improving the generalization error for both the target languages.

Next, we analyze the results of *one-to-many* translation experiments when both the target languages belong to distant language families and are unrelated. The first set of experiments are on *Germanic, Turkic* languages (En→De+Tr) and the second set of experiments are on *Germanic, Japonic* languages (En→De+Ja). We present the results in Table 2b when different sets of parameters are shared. Here, we observe that the approach of sharing all the parameters leads to a noticeable drop in the BLEU scores for both the considered language pairs. Similar to the above discussion, sharing the key and query matrices results in a large increase in the BLEU scores. We hypothesize that in this partial parameter sharing strategy, the sharing of key and query attention weights effectively models the common linguistic properties while the separate FFN sublayer weights model the unique characteristics of each target language, thus overall leading to a large improvement in the BLEU scores. The results of other decoder parameter sharing approaches lie close to the key and query parameter sharing method. As the target languages are from different families, their vocabularies may have some overlap but will be significantly different from each other. In this scenario, a useful alternative is to consider a separate embedding layer for every source-target language pair while sharing all the encoder and decoder parameters. However, we did not experiment with this approach, as the inclusion of separate embedding layers will lead to a large increase in the model parameters and as a result model training will become more memory intensive. We leave the investigation of such parameter sharing strategy to future work.

### 4.2 Overall Comparison

In Table 3, we show an overall performance comparison of no parameter sharing, full parameter sharing for both GNMT (Wu et al., 2016) and Transformer models, and the best approaches according to maximum BLEU score from our partial parameter sharing strategies. For training the GNMT models, we use its open-source implementation[4] (Luong et al., 2017) with four layers[5] and default parameter settings. First, we note that the BLEU scores of the Transformer model are always better than the GNMT model by a significant margin for both bilingual (no sharing) and multilingual

[4] https://github.com/tensorflow/nmt

[5] We found that the four layer model for GNMT didn't overfit and obtained the best BLEU scores.

| *Set of shared parameters* ($\Theta$) | En→Ro+Fr | | En→De+Nl | | *params* |
|---|---|---|---|---|---|
| | →Ro | →Fr | →De | →Nl | $\times 10^6$ |
| $W_E$ | 27.21 | 43.36 | 30.32 | 33.51 | 105 |
| $W_E$, $\theta_{ENC}$ | 27.82 | 43.83 | 29.97 | 33.33 | 86 |
| $W_E$, $\theta_{ENC}$, $W_1$, $W_2$ | 27.78 | 43.87 | 29.95 | 33.12 | 74 |
| $W_E$, $\theta_{ENC}$, $W_K^1$, $W_Q^1$, $W_V^1$, $W_F^1$ | 27.80 | 43.76 | 30.68 | 33.99 | 80 |
| $W_E$, $\theta_{ENC}$, $W_K^2$, $W_Q^2$, $W_V^2$, $W_F^2$ | 28.36 | 44.19 | 30.50 | 33.75 | 80 |
| $W_E$, $\theta_{ENC}$, $W_K^1$, $W_V^1$, $W_K^2$, $W_V^2$ | 27.77 | 43.83 | 30.54 | 34.00 | 80 |
| $W_E$, $\theta_{ENC}$, $W_K^1$, $W_Q^1$, $W_K^2$, $W_Q^2$ | 27.58 | 43.84 | **30.70** | **34.05** | 80 |
| $W_E$, $\theta_{ENC}$, $W_K^1$, $W_Q^1$, $W_V^1$, $W_F^1$, $W_K^2$, $W_Q^2$, $W_V^2$, $W_F^2$ | 28.14 | 44.12 | 30.64 | 33.92 | 74 |
| $W_E$, $\theta_{ENC}$, $\theta_{DEC}$ | **28.52** | **44.28** | 30.45 | 33.69 | 61 |

**(a)** The target languages in this *one-to-many* translation task belong to the same language family. Ro and Fr are *Romance* languages while De and Nl are *Germanic* languages.

| *Set of shared parameters* ($\Theta$) | En→De+Tr | | En→De+Ja | | *params* |
|---|---|---|---|---|---|
| | →De | →Tr | →De | →Ja | $\times 10^6$ |
| $W_E$ | 30.35 | 19.66 | 30.10 | 18.62 | 105 |
| $W_E$, $\theta_{ENC}$ | 30.55 | 19.29 | 30.21 | 18.70 | 86 |
| $W_E$, $\theta_{ENC}$, $W_{L_1}$, $W_{L_2}$ | 30.21 | 19.17 | 30.36 | 18.92 | 74 |
| $W_E$, $\theta_{ENC}$, $W_K^1$, $W_Q^1$, $W_V^1$, $W_F^1$ | 30.35 | 19.24 | 30.05 | 18.78 | 80 |
| $W_E$, $\theta_{ENC}$, $W_K^2$, $W_Q^2$, $W_V^2$, $W_F^2$ | 30.49 | 19.40 | 30.16 | 18.73 | 80 |
| $W_E$, $\theta_{ENC}$, $W_K^1$, $W_V^1$, $W_K^2$, $W_V^2$ | 30.66 | 19.34 | 30.36 | 18.92 | 80 |
| $W_E$, $\theta_{ENC}$, $W_K^1$, $W_Q^1$, $W_K^2$, $W_Q^2$ | **30.71** | **19.67** | **30.48** | **19.00** | 80 |
| $W_E$, $\theta_{ENC}$, $W_K^1$, $W_Q^1$, $W_V^1$, $W_F^1$, $W_K^2$, $W_Q^2$, $W_V^2$, $W_F^2$ | 30.40 | 19.35 | 30.35 | 18.80 | 74 |
| $W_E$, $\theta_{ENC}$, $\theta_{DEC}$ | 28.74 | 18.69 | 29.68 | 18.50 | 61 |

**(b)** The target languages in this *one-to-many* translation task belong to distant language families. De, Tr, and Ja are unrelated as they belong to *Germanic*, *Turkic*, and *Japonic* language families respectively.

**Table 2:** BLEU scores for various parameter sharing strategies when the target languages either belong to the same family ({Ro, Fr}, {De, Nl}) or to distant families (De, Tr, Ja). $\theta_{ENC}$ denotes that all the encoder parameters are shared between the models; $\theta_{DEC}$ denotes that all the decoder parameters are shared between the models.

| Method | En→De+Tr | | En→De+Ja | | En→Ro+Fr | | En→De+Nl | | *params* |
|---|---|---|---|---|---|---|---|---|---|
| | →De | →Tr | →De | →Ja | →Ro | →Fr | →De | →Nl | $\times 10^6$ |
| GNMT NS | 27.01 | 16.07 | 27.01 | 16.62 | 24.38 | 40.50 | 27.01 | 30.64 | – |
| GNMT FS | 29.07 | 18.09 | 28.24 | 17.33 | 26.41 | 42.46 | 28.52 | 31.72 | – |
| Transformer NS | 29.31 | 18.62 | 29.31 | 17.92 | 26.81 | 42.95 | 29.31 | 32.43 | 122 |
| Transformer FS | 28.74 | 18.69 | 29.68 | 18.50 | **28.52** | **44.28** | 30.45 | 33.69 | 61 |
| Transformer PS | **30.71** | **19.67** | **30.48** | **19.00** | 27.58 | 43.84 | **30.70** | **34.05** | 80 |

**Table 3:** BLEU scores for different models for *one-to-many* translation task. **NS**: *No Sharing* corresponds to the bilingual models when the two language pairs are trained independently; **FS**: *Full Sharing* means one model is used for the translation of all the language pairs; **PS**: *Partial Sharing* means that the embedding, encoder, decoder's key, and value weights are shared between the two models.

(full sharing) translation tasks. This reflects that the Transformer model is well-suited for both multilingual and bilingual translation tasks compared with the GNMT model. We also surprisingly note that the GNMT fully shared model is able to consistently obtain higher BLEU scores compared with its bilingual version irrespective of which families the target languages belong to.

However, for the *one-to-many* translation task when the target languages are from distant families, we observe that fully shared Transformer model leads to a substantial drop or small gains in the BLEU score compared with the bilingual models. Specifically, for the EN→DE+TR setting, BLEU drops by 0.6 for EN→DE, while staying even for EN→TR. In contrast, our method of sharing embedding, encoder, decoder's key, and query parameters leads to substantial increases in BLEU scores (1.4↑ for EN→DE and 1.1↑ for EN→TR). Similarly, for EN→DE+JA, using the fully shared Transformer model, we observe small gains of 0.3 and 0.5 BLEU points for EN→DE and EN→JA respectively while our partial parameter sharing method again leads to significant improvements (1.5↑ for EN→DE and 1.1↑ for EN→JA). This demonstrates the utility of our proposed partial parameter sharing method.

We also note that fully shared Transformer models can be an effective strategy only when both the target languages are from the same family. For the task of EN→RO+FR, the fully shared model performs surprisingly well and yields significant improvements of 1.7 and 1.3 BLEU points compared with bilingual models for EN→RO and EN→FR respectively. A similar increase in performance can also be observed for the EN→DE+NL task, although for this task, our partial parameter sharing method (encoder, embedding, decoder's key, and query weights) obtains even higher BLEU scores. (1.4↑ for EN→DE and 1.6↑ EN→NL).

### 4.3 Analysis

Here, we analyze the generated translations of the partial sharing and full sharing approaches for EN→DE when *one-to-many* multilingual model was trained on unrelated target language pairs EN→DE+TR. These translations were obtained using the test set of EN→DE task. Here partial sharing refers to the specific approach of sharing the embedding, encoder, and decoder's key and query parameters in the model.

We show example translations in Table 4 where partial sharing method gets a high BLEU score (shown in parentheses) but the full sharing method does not. We see that sentences generated by partial sharing method are both semantically and grammatically correct while the full sharing method generates shorter sentences compared with reference translations. As highlighted in table cells, the partial sharing method is able to correctly translate a mention of relative time "*half a year*" and a co-reference expression "*mich*". In contrast, the fully shared model generates incorrect expressions of time mentions "*eineinhalb Jahren*" (one and half years) and different verb forms ("*schlägt*" is generated vs "*schlagen*" in the reference).

We also perform a comparison of the F-measure of the target words for EN→DE, bucketed by frequency in the training set. As displayed in Figure 4, this shows that the partial parameter sharing approach improves the translation accuracy for the entire vocabulary, but in particular for words that have low-frequency in the dataset.

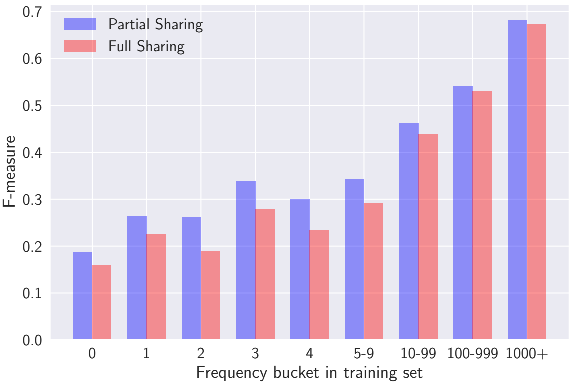


**Figure 4:** The F-measure for the target language (DE) words in *one-to-many* multilingual translation task (EN→DE+TR). Best viewed in color.

## 5 Related Work

In this section, we will review the prior work related to MTL and multilingual translation.

### 5.1 Multi-task learning

Ando and Zhang (2005) obtained excellent results by adopting an MTL framework to jointly train linear models for NER, POS tagging, and language modeling tasks involving some degree of parameter sharing. Later, Collobert et al. (2011) applied MTL strategies to neural networks for tasks such as POS tagging, NER, and chunking by sharing the

| | |
|---|---|
| **source** | So half a year ago , I decided to go to Pakistan myself . |
| **reference** | Vor einem halben Jahr entschied ich mich , selbst nach Pakistan zu gehen . |
| **partial sharing** | Vor einem halben Jahr entschied ich mich , selbst nach Pakistan zu gehen . (1.0) |
| **full sharing** | Vor eineinhalb Jahren beschloss ich , nach Pakistan zu gehen . (0.35) |
| **source** | Your heart starts beating faster . |
| **reference** | Ihr Herz beginnt schneller zu schlagen . |
| **partial sharing** | Ihr Herz beginnt schneller zu schlagen . (1.0) |
| **full sharing** | Ihr Herz schlägt schneller . (0.27) |

**Table 4:** Sample translations from En→De when *one-to-many* multilingual model was trained on unrelated target language pairs En→De+Tr. In these examples, the method of partial sharing of decoder parameters obtains a very high BLEU score (mentioned in parentheses).

sequence encoder and reported moderate improvements in results. Recently, Luong et al. (2016) investigated MTL for a tasks such as parsing, image captioning, and translation and observed large gains in the translation task. Similarly, for MT tasks, Niehues and Cho (2017) also leverage MTL by using additional linguistic information to improve the translation accuracy of NMT models. They share the encoder representations to perform joint training on translation, POS, and NER tasks. MTL has also been widely applied to multilingual translation that will be discussed next.

### 5.2 Multilingual Translation

On the multilingual translation task, Dong et al. (2015) obtained significant performance gains by sharing the encoder parameters of the source language while having a separate decoder for each target language. Later, Firat et al. (2016) attempted the more challenging task of *many-to-many* translation by training a model that consisted of one shared encoder and decoder per language and a shared attention layer that was common to all languages. This approach obtained competitive BLEU scores on ten European language pairs while substantially reducing the total parameters. Recently, Johnson et al. (2017) proposed a unified model with full parameter sharing and obtained comparable or better performance compared with bilingual translation scores. During model training and decoding, target language was specified by an additional token at the beginning of the source sentence. Coming to low-resource language translation, Zoph et al. (2016) used a transfer learning approach of fine-tuning the model parameters learned on a high-resource language pair of French→English and were able to significantly increase the translation performance on Turkish and Urdu languages. Recently, Gu et al. (2018) addresses the *many-to-one* translation problem for extremely low-resource languages by using a transfer learning approach such that all language pairs share the lexical and sentence-level representations. By performing joint training of the model with high-resource languages, large gains in the BLEU scores were reported for low-resource languages.

In this paper, we first experiment with the Transformer model for *one-to-many* multilingual translation on a variety of language pairs and demonstrate that the approach of Johnson et al. (2017) and Dong et al. (2015) is not optimal for all kinds of target-side languages. Motivated by this, we introduce various parameter sharing strategies that strike a happy medium between full sharing and partial sharing and show that it achieves the best translation accuracy.

## 6 Conclusion

In this work, we explore parameter sharing strategies for the task of multilingual machine translation using self-attentional MT models. Specifically, we examine the case when the target languages come from the same or distant language families. We show that the popular approach of full parameter sharing may perform well only when the target languages belong to the same family while a partial parameter sharing approach consisting of shared embedding, encoder, decoder's key and query weights is generally applicable to all kinds of language pairs and achieves the best BLEU scores when the languages are from distant families.

For future work, we plan to extend our parameter sharing approach in two directions. First, we aim to increase the number of target languages to more than two such that they contain a mix of both similar and distant languages and analyze the performance of our proposed parameter sharing strategies on them. Second, we aim to experiment with additional parameter sharing strategies such as sharing the weights of some specific layers (*e.g.* the first or last layer) as different layers can encode different morphological information (Belinkov et al., 2017) which can be helpful in better multilingual translation.

## Acknowledgments

The authors would like to thank Mrinmaya Sachan, Emmanouil Antonios Platanios, and Soumya Wadhwa for useful comments about this work. We would also like to thank the anonymous reviewers for giving us their valuable feedback that helped to improve the paper.

## References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal Machine Learning Research*, 6:1817–1853.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *Computing Research Repository*, arXiv:1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *Computing Research Repository*, arXiv:1409.0473.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. 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)*, pages 1724–1734, Doha, Qatar.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354.

Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2016. Tying word vectors and word classifiers: A loss framework for language modeling. *Computing Research Repository*, arXiv:1611.01462.

Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computing Research Repository*, arXiv:1412.6980.

Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 1998. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer.

Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. 2017. Neural machine translation (seq2seq) tutorial. *https://github.com/tensorflow/nmt*.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA.

Jan Niehues and Eunah Cho. 2017. Exploiting linguistic resources for neural machine translation using multi-task learning. In *Proceedings of the Second Conference on Machine Translation*, pages 80–89.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *Computing Research Repository*, arXiv:1701.06548.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *Computing Research Repository*, arXiv:1609.08144.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575. Association for Computational Linguistics.

# Findings of the 2018 Conference on Machine Translation (WMT18)

**Ondřej Bojar**
Charles University

**Christian Federmann**
Microsoft Cloud + AI

**Mark Fishel**
University of Tartu

**Yvette Graham**
Dublin City University

**Barry Haddow**
University of Edinburgh

**Matthias Huck**
LMU Munich

**Philipp Koehn**
JHU / University of Edinburgh

**Christof Monz**
University of Amsterdam

## Abstract

This paper presents the results of the premier shared task organized alongside the Conference on Machine Translation (WMT) 2018. Participants were asked to build machine translation systems for any of 7 language pairs in both directions, to be evaluated on a test set of news stories. The main metric for this task is human judgment of translation quality. This year, we also opened up the task to additional test sets to probe specific aspects of translation.

## 1 Introduction

The Third Conference on Machine Translation (WMT) held at EMNLP 2018[1] host a number of shared tasks on various aspects of machine translation. This conference builds on twelve previous editions of WMT as workshops and conferences (Koehn and Monz, 2006; Callison-Burch et al., 2007, 2008, 2009, 2010, 2011, 2012; Bojar et al., 2013, 2014, 2015, 2016a, 2017).

This year we conducted several official tasks. We report in this paper on the news translation task. Additional shared tasks are described in separate papers in these proceedings:

- biomedical translation (Neves et al., 2018),
- multimodal machine translation (Barrault et al., 2018),
- metrics (Ma et al., 2018),
- quality estimation (Specia et al., 2018),
- automatic post-editing (Chatterjee et al., 2018), and
- parallel corpus filtering (Koehn et al., 2018b).

In the news translation task (Section 2), participants were asked to translate a shared test set, optionally restricting themselves to the provided training data ("constrained" condition). We held 14 translation tasks this year, between English and each of Chinese, Czech, Estonian, German, Finnish, Russian, and Turkish. The Estonian-English language pair was new this year. Similarly to Latvian, which we had covered in 2017, Estonian is a lesser resourced data condition on a challenging language pair. System outputs for each task were evaluated both automatically and manually.

This year the news translation task had two additional sub-tracks: multilingual MT and unsupervised MT. Both sub-tracks were included into the general list of news translation submissions and are described in more detail in corresponding subsections of Section 2.

The human evaluation (Section 3) involves asking human judges to score sentences output by anonymized systems. We obtained large numbers of assessments from researchers who contributed evaluations proportional to the number of tasks they entered. In addition, we used Mechanical Turk to collect further evaluations. This year, the official manual evaluation metric is again based on judgments of adequacy on a 100-point scale, a method we explored in the previous years with convincing results in terms of the trade-off between annotation effort and reliable distinctions between systems.

The primary objectives of WMT are to evaluate the state of the art in machine translation, to disseminate common test sets and public training data with published performance numbers, and to refine evaluation and estimation methodologies for machine translation. As before, all of the data, translations, and collected human judgments are publicly available[2]. We hope these datasets serve as a valuable resource for research into data-driven machine translation, automatic evaluation, or prediction of translation quality. News transla-

[1] http://www.statmt.org/wmt18/

[2] http://statmt.org/wmt18/results.html

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 272–303
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64028

tions are also available for interactive visualization and comparison of differences between systems at http://wmt.ufal.cz/ using MT-ComparEval (Sudarikov et al., 2016).

In order to gain further insight into the performance of individual MT systems, we organized a call for dedicated "test suites", each focussing on some particular aspect of translation quality. A brief overview of the test suites is provided in Section 4.

## 2 News Translation Task

The recurring WMT task examines translation between English and other languages in the news domain. As in the previous year, we include Chinese, Czech, German, Finnish, Russian, and Turkish. A new language this year is Estonian.

We created a test set for each language pair by translating newspaper articles and provided training data.

### 2.1 Test Data

The test data for this year's task was selected from online sources, as in previous years. We took about 1500 English sentences and translated them into the other languages, and then additional 1500 sentences from each of the other languages and translated them into English. This gave us test sets of about 3000 sentences for our English-X language pairs, which have been either originally written in English and translated into X, or vice versa. The composition of the test documents is shown in Table 1, the size of the test sets in terms is given in Figure 2.

The stories were translated by professional translators, funded by the EU Horizon 2020 projects CRACKER and QT21 (German, Czech), by Yandex[3], a Russian search engine company (Turkish, Russian), by BAULT, a research community on building and using language technology funded by the University of Helsinki (Finnish) and the University of Tartu[4] (Estonian). The Chinese–English task was sponsored by Nanjing University, Xiamen University, the Institutes of Computing Technology and of Automation, Chinese Academy of Science, Northeastern University (China) and Datum Data Co., Ltd. All of the translations were done directly, and not via an intermediate language.

Since Estonian-English was run for the first time, both the test and development set had to be translated: the size of both was 2000 sentences (4000 in total).

### 2.2 Training Data

As in past years we provided parallel corpora to train translation models, monolingual corpora to train language models, and development sets to tune system parameters. Some training corpora were identical from last year (Europarl,[5] Common Crawl, SETIMES2 , Russian-English parallel data provided by Yandex, Wikipedia Headlines provided by CMU) and some were updated (United Nations, CzEng v1.7 (Bojar et al., 2016b), News Commentary v13, monolingual news data). A new corpus is the EU Press Release parallel corpus for German, Finnish, and Latvian.

For Latvian and Chinese, a number of new corpora were released. For Latvian, this data was prepared by the University of Latvia and Tilde, the Chinese corpora were prepared by the Institutes of Computing Technology and of Automation, Chinese Academy of Science, Northeastern University (China) and Datum Data Co., Ltd.

Some statistics about the training materials are given in Figure 1.

### 2.3 Submitted Systems

We received 103 submissions from 32 institutions. The participating institutions, organized into 35 teams are listed in Table 2 and detailed in the rest of this section. Each system did not necessarily appear in all translation tasks. We also included 39 online MT systems (originating from 5 services), which we anonymized as ONLINE-A,B,F,G.

For presentation of the results, systems are treated as either *constrained* or *unconstrained*, depending on whether their models were trained only on the provided data. Since we do not know how they were built, these online and commercial systems are treated as unconstrained during the automatic and human evaluations.

#### 2.3.1 AALTO (Grönroos et al., 2018)

Aalto participated in the constrained condition of the multi-lingual subtrack, with a single system trained to translate from English to both Finnish

[3] http://www.yandex.com/

[4] Estonian Research Council institutional research grant IUT20-56: "Computational models of the Estonian Language"

[5] As of Fall 2011, the proceedings of the European Parliament are no longer translated into all official languages.

## Europarl Parallel Corpus

| | German ↔ English | | Czech ↔ English | | Finnish ↔ English | | Estonian ↔ English | |
|---|---|---|---|---|---|---|---|---|
| **Sentences** | 1,920,209 | | 646,605 | | 1,926,114 | | 652,944 | |
| **Words** | 50,486,398 | 53,008,851 | 14,946,399 | 17,376,433 | 37,814,266 | 52,723,296 | 13,033,918 | 17,453,613 |
| **Distinct words** | 381,583 | 115,966 | 172,461 | 63,039 | 693,963 | 115,896 | 298,021 | 63,432 |

## News Commentary Parallel Corpus

| | German ↔ English | | Czech ↔ English | | Russian ↔ English | | Chinese ↔ English | |
|---|---|---|---|---|---|---|---|---|
| **Sentences** | 284,246 | | 218,384 | | 235,159 | | 252,777 | |
| **Words** | 7,243,776 | 7,174,644 | 4,942,255 | 5,411,117 | 6,230,738 | 6,230,738 | – | 6,428,459 |
| **Distinct words** | 182,059 | 75,590 | 166,173 | 66,054 | 71,021 | 71,021 | – | 70,092 |

## Common Crawl Parallel Corpus

| | German ↔ English | | Czech ↔ English | | Russian ↔ English | |
|---|---|---|---|---|---|---|
| **Sentences** | 2,399,123 | | 161,838 | | 878,386 | |
| **Words** | 54,575,405 | 58,870,638 | 3,529,783 | 3,927,378 | 21,018,793 | 21,535,122 |
| **Distinct words** | 1,640,835 | 823,480 | 210,170 | 128,212 | 764,203 | 432,062 |

## ParaCrawl Parallel Corpus

| | German ↔ English | | Czech ↔ English | | Estonian ↔ English | |
|---|---|---|---|---|---|---|
| **Sentences** | 36,351,593 | | 10,020,250 | | 1,298,103 | |
| **Words** | 595,027,749 | 623,361,284 | 116,797,931 | 122,699,058 | 37,887,435 | 39,060,095 |
| **Distinct Words** | 8065519 | 5,371,211 | 1,912,633 | 1,538,696 | 1,025,961 | 894,357 |

| | Finnish ↔ English | | Russian ↔ English | |
|---|---|---|---|---|
| **Sentences** | 624,058 | | 1,2061,155 | |
| **Words** | 8,636,936 | 11,123,014 | 182,229,052 | 210,751,004 |
| **Distinct Words** | 379,958 | 127,006 | 3,164,200 | 2,415,633 |

## EU Press Release Parallel Corpus

| | German ↔ English | | Finnish ↔ English | | Estonian ↔ English | |
|---|---|---|---|---|---|---|
| **Sentences** | 1,329,041 | | 583,223 | | 226978 | |
| **Words** | 25,048,312 | 25,777,997 | 8,052,607 | 11,244,602 | 3,940,058 | 177,723 |
| **Distinct words** | 398,477 | 168,725 | 315,394 | 94,979 | 5,209,544 | 57,059 |

## Chinese Parallel Corpora

| | casia2015 | casict2011 | casict2015 | datum2011 | datum2017 | neu2017 |
|---|---|---|---|---|---|---|
| **Sentences** | 1,050,000 | 1,936,633 | 2,036,834 | 1,000,004 | 999,985 | 2,000,000 |
| **Words (en)** | 20,571,578 | 34,866,598 | 22,802,353 | 24,632,984 | 25,182,185 | 29,696,442 |
| **Distinct words (en)** | 470,452 | 627,630 | 435,010 | 316,277 | 312,164 | 624,420 |

## Yandex 1M Parallel Corpus

| | Russian ↔ English | |
|---|---|---|
| **Sentences** | 1,000,000 | |
| **Words** | 24,121,459 | 26,107,293 |
| **Distinct** | 701,809 | 387,646 |

## Wiki Headlines Parallel Corpus

| | Russian ↔ English | | Finnish ↔ English | |
|---|---|---|---|---|
| **Sentences** | 514,859 | | 153,728 | |
| **Words** | 1,191,474 | 1,230,644 | 269,429 | 354,362 |
| **Distinct** | 282,989 | 251,328 | 127,576 | 96,732 |

## CzEng v1.7 Parallel Corpus

| | Czech ↔ English | |
|---|---|---|
| **Sentences** | 61,243,252 | |
| **Words** | 737,434,097 | 3,650,518 |
| **Distinct** | 835,192,627 | 2,580,902 |

## SE Times 2 Parallel Corpus

| | Turkish ↔ English | |
|---|---|---|
| **Sentences** | 207,678 | |
| **Words** | 4,626,277 | 5,147,769 |
| **Distinct** | 155,479 | 69,927 |

## United Nations Parallel Corpus

| | Russian ↔ English | | Chinese ↔ English | |
|---|---|---|---|---|
| **Sentences** | 23,239,280 | | 15,886,041 | |
| **Words** | 482,966,738 | 524,719,646 | – | 372,612,596 |
| **Distinct** | 3,857,656 | 2,737,469 | – | 1,981,413 |

**Figure 1:** Statistics for the training sets used in the translation task. The number of words and the number of distinct words (case-insensitive) is based on the provided tokenizer.

| Language | Sources (Number of Documents) |
|---|---|
| **English** | ABC News (1), BBC (4), Brisbane Times (1), CBS News (1), Daily Mail (4), Euronews (3), Globe and Mail (1), Guardian (4), Independent (4), Los Angeles Times (4), MSNBC (3), Novinte (2), New York Times (2), Reuters (3), Russia Today (2), Scotsman (2), Sydney Morning Herald (2), Telegraph (2), The Local (2), Time Magazine (2), UPI (1), Washington Post (3) |
| **Czech** | blesk.cz (16), deník.cz (5), Deník Referendum (1), DNES.cz (7), lidovky.cz (6), Novinky.cz (3), Reflex (2), tyden.cz (12), ZDN (2) |
| **German** | Aachener Nachrichten (1), Abendzeitung Nürnberg (2), Braunschweiger Zeitung (1), Der Standard (1), Die Presse (1), Euronews (1), Fehmarn24 (1), Handelsblatt (1), Hannoversche Allgemeine (2), Hessische/Niedersächsische Allgemeine (1), In Franken (4), Kreiszeitung (2), Krone (1), Mainpost (1), Merkur (3), Morgenpost (1), n-tv (1), Neue Westfälische (1), oe24 (2), Peiner Allgemeine (1), Passauer Neue Presse (2), Rheinzeitung (1), Rundschau (1), Schwarzwälder Bote (16), Segeberger Zeitung (2), Südkurier (1), Thüringer Allgemeine (1), Thüringer Landeszeitung (1), Volksblatt (2), Volksfreund (3), Westfälische Nachrichten (1), Westdeutsche Zeitung (8). |
| **Estonian** | Arileht (7), Maaleht (3), Postimees (17), Sloleht (23). |
| **Finnish** | Etelä-Saimaa (2), Etelä-Suomen Sanomat (3), Helsingin Sanomat (4), Iltalehti (13), Ilta-Sanomat (29), Kaleva (12), Kansan Uutiset (1), Karjalainen (13), Kouvolan Sanomat (2). |
| **Russian** | aif (4), Altapress (1), Argumenti (19), ERR.ee (3), eg-online.ru (2), Euronews (2), Fakty (5), Infox (2), Izvestiya (25), Kommersant (16), Lenta (9), lgng (3), MK RU (5), nov-pravda.ru (1), pnp.ru (6), rg.ru (4), Vedomosti (3), Versia (1), Vesti (3), zr.ru (1) |
| **Turkish** | Hürriyet.com (48), Sabah (96), Sözcü (19) |

**Table 1:** Composition of the test set. For more details see the XML test files. The `docid` tag gives the source and the date for each document in the test set, and the `origlang` tag indicates the original source language.

**BigEst Estonian Corpus**

| | |
|---|---|
| **Sentences** | 40,404,948 |
| **Words** | 579,221,489 |
| **Distinct words** | 8,134,555 |

**News Language Model Data**

| | English | German | Czech | Russian | Finnish | Turkish | Estonian |
|---|---|---|---|---|---|---|---|
| **Sentences** | 192,988,741 | 260,754,881 | 66,517,569 | 39,519,008 | 14,575,981 | 4,753,928 | 817,472 |
| **Words** | 4,428,839,473 | 4,627,780,738 | 1,094,215,341 | 724,582,848 | 184,523,981 | 79,067,739 | 12,880,832 |
| **Distinct words** | 6,468,049 | 20,276,165 | 4,269,005 | 3,397,828 | 4,391,543 | 1,025,791 | 653,980 |

**Common Crawl Language Model Data**

| | English | German | Czech | Russian | Finnish | Estonian | Turkish | Chinese |
|---|---|---|---|---|---|---|---|---|
| **Sent.** | 3,074,921,453 | 2,872,785,485 | 333,498,145 | 1,168,529,851 | 157,264,161 | 100,779,314 | 511,196,951 | 1,672,324,647 |
| **Words** | 65,128,419,540 | 65,154,042,103 | 6,694,811,063 | 23,313,060,950 | 2,935,402,545 | 2,906,100,138 | 11,882,126,872 | – |
| **Dist.** | 342,760,462 | 339,983,035 | 50,162,437 | 101,436,673 | 47,083,545 | 27,618,190 | 88,463,295 | – |

**Test Set**

| | Czech ↔ EN | | German ↔ EN | | Finnish ↔ EN | | Estonian ↔ EN | |
|---|---|---|---|---|---|---|---|---|
| **Sentences.** | 2983 | | 2998 | | 3000 | | 2000 | |
| **Words** | 47,229 | 55,920 | 54,933 | 58,628 | 38,149 | 54,790 | 30,531 | 40,158 |
| **Distinct words** | 18,325 | 12,548 | 15,996 | 13,431 | 17,825 | 12,043 | 14,185 | 10,096 |

| | Russian ↔ EN | | Turkish ↔ EN | | Chinese ↔ EN | |
|---|---|---|---|---|---|---|
| **Sentences.** | 3000 | | 3000 | | 3981 | |
| **Words** | 51,988 | 62,925 | 45,944 | 60,232 | – | 98,308 |
| **Distinct words** | 21,116 | 13,584 | 19,200 | 13,444 | – | 16,955 |

**Figure 2:** Statistics for the training and test sets used in the translation task. The number of words and the number of distinct words (case-insensitive) is based on the provided tokenizer.

| Team | Institution |
|---|---|
| AALTO | Aalto University (Grönroos et al., 2018) |
| AFRL | Air Force Research Laboratory (Gwinnup et al., 2018) |
| ALIBABA | Alibaba Group (Deng et al., 2018) |
| CUNI-KOCMI | Charles University (Kocmi et al., 2018) |
| CUNI-TRANSFORMER | Charles University (Popel, 2018) |
| FACEBOOK-FAIR ⋆ | Facebook AI Research (Edunov et al., 2018) |
| GTCOM | Global Tone Communication Technology (Bei et al., 2018) |
| HY | University of Helsinki (Raganato et al., 2018) |
| JHU | Johns Hopkins University (Koehn et al., 2018a) |
| JUCBNMT | Jadavpur University (Mahata et al., 2018) |
| KIT | Karlsruhe Institute of Technology (Pham et al., 2018) |
| LI-MUZE | Li Muze (no associated paper) |
| LMU-NMT | LMU Munich (Huck et al., 2018) |
| LMU-UNSUP | LMU Munich (Stojanovski et al., 2018) |
| MICROSOFT-MARIAN | Microsoft (Junczys-Dowmunt, 2018) |
| MLLP-UPV | MLLP, Technical University of Valencia (Iranzo-Sánchez et al., 2018) |
| MMT-PRODUCTION | ModernMT, MMT s.r.l. (no associated paper) |
| NEUROTOLGE.EE | University of Tartu (Tars and Fishel, 2018) |
| NICT | National Institute of Information and Communications Technology (Marie et al., 2018) |
| NIUTRANS | Northeastern University / NiuTrans Co., Ltd. (Wang et al., 2018b) |
| NJUNMT | NLP Group, Nanjing University (no associated paper) |
| NTT | NTT Corporation (Morishita et al., 2018) |
| PARFDA | Boğaziçi University (Biçici, 2018) |
| PROMT | PROMT LLC (Molchanov, 2018) |
| RWTH | RWTH Aachen (Schamper et al., 2018) |
| RWTH-UNSUPER | RWTH Aachen (Graça et al., 2018) |
| TALP-UPC | TALP, Technical University of Catalonia (Casas et al., 2018) |
| TENCENT | Tencent (Wang et al., 2018a) |
| TILDE | Tilde (Pinnis et al., 2018) |
| UBIQUS | Ubiqus (no associated paper) |
| UCAM | University of Cambridge (Stahlberg et al., 2018) |
| UEDIN | University of Edinburgh (Haddow et al., 2018) |
| UMD | University of Maryland (Xu and Carpuat, 2018) |
| UNISOUND | Unisound (no associated paper) |
| UNSUPTARTU | University of Tartu (Del et al., 2018) |

**Table 2:** Participants in the shared translation task. Not all teams participated in all language pairs. The translations from the commercial and online systems were not submitted by their respective companies but were obtained by us, and are therefore anonymized in a fashion consistent with previous years of the workshop. "⋆" indicates invited participation with a late submission, where the team is not considered a regular participant.

and Estonian. The system is based on the Transformer (Vaswani et al., 2017) implementation in OpenNMT-py (Klein et al., 2017). It is trained on filtered parallel and filtered back-translated monolingual data. The main contribution is a novel cross-lingual Morfessor (Virpioja et al., 2013) segmentation using cognates extracted from the parallel data. The aim is to improve the consistency of the morphological segmentation. Aalto decode using an ensemble of 3 (et) or 8 (fi) models.

#### 2.3.2 AFRL (Gwinnup et al., 2018)

AFRL-SYSCOMB is a system-combination entry consisting of three inputs. The first is an OpenNMT system trained on the provided parallel data except ParaCrawl and the backtranslated corpus used in the AFRL WMT17 system (Gwinnup et al., 2017). This system uses a standard RNN architecture and was fine-tuned with the other available news task test sets. The second is a Marian (Junczys-Dowmunt et al., 2018) system ensembling 5 Univ. Edinburgh "bi-deep" and 6 transformer models all trained on the WMT18 bitexts provided, including ParaCrawl. Some models employed pretrained word embeddings built on BPE'd corpora (Sennrich et al., 2016). A Marian transformer model performed right-to-left rescoring for this system. The third system is trained with Moses (Koehn et al., 2007), using the same data as the Marian system. Hierarchical reordering and Operation Sequence Model were employed. The 5-gram English language model was trained with KenLM (Heafield, 2011) on the same corpus as the AFRL WMT15 system with the same BPE used in the Marian systems. Lastly, RWTH Jane's system combination (Freitag et al., 2014) was applied yielding approximately a +0.5 gain in BLEU.

#### 2.3.3 ALIBABA (Deng et al., 2018)

Alibaba systems are based on the Transformer model architecture, with the most recent features from the academic research integrated, such as weighted Transformer, Transformer with relative position attention, etc. The system also employs most techniques that have been proven effective during the past WMT years, such as BPE-based subword, back translation, fine-tuning based on selected data, model ensembling and reranking, at industrial scale. For some morphologically-rich languages, linguistic knowledge is also incorporated into the neural network.

#### 2.3.4 CUNI-KOCMI (Kocmi et al., 2018)

The CUNI-KOCMI submission focuses on the low-resource language neural machine translation (NMT). The final submission uses a method of transfer learning: the model is pretrained on a related high-resource language (here Finnish) first, followed by a child low-resource language (Estonian) without any change in hyperparameters. Averaging and backtranslation are also experimented with.

#### 2.3.5 CUNI-TRANSFORMER (Popel, 2018)

CUNI-TRANSFORMER is the Transformer model trained according to Popel and Bojar (2018) plus a novel concat-regime backtranslation with checkpoint averaging, tuned separately for CZ-domain and nonCZ-domain articles, possibly handling also translation-direction ("translationese") issues. For cs→en also a coreference preprocessing was used adding the female-gender pronoun where it was pro-dropped in Czech, referring to a human and could not be inferred from a given sentence.

#### 2.3.6 FACEBOOK-FAIR ⋆ (Edunov et al., 2018)

FACEBOOK-FAIR is an ensemble of six self-attentional models with back-translation data according to Edunov et al. (2018). Synthetic sources are sampled instead of beam search, oversampling the real bitext at a rate of 16, i.e., each bitext is sampled 16 times more often per epoch than the back-translated data. At inference time, translations which are copies of the source are filtered out, replacing them with the output of a very small news-commentary only trained model.

The system FACEBOOK-FAIR has been submitted anonymously as ONLINE-Z and approval for disclosing the authors' identity has only been granted after the final results had become available. Due to the non-standard way of submission, the system is not considered a regular participant, but an invited/late submission and marked with "⋆" throughout the paper.

#### 2.3.7 GTCOM (Bei et al., 2018)

GTCOM-PRIMARY is based on the Transformer "base" model architecture using Marian toolkit, and it also applies some methods that have been proven effective in NMT system, such as BPE, back-translation, right-to-left reranking and ensembling decoding. In this experiment, right-to-left reranking does not help. Another focus is given to data filtering through rules, translation model and language model including parallel data and monolingual data. The language model is based the Transformer architecture as well. The final system is trained with four different seeds and mixed data.

### 2.3.8 HY (Raganato et al., 2018; Hurskainen and Tiedemann, 2017)

The University of Helsinki (HY) submitted four systems: HY-AH, HY-NMT, HY-NMT-2STEP and HY-SMT.

HY-AH (**Raganato et al., 2018; Hurskainen and Tiedemann, 2017**) is a rule-based machine translation system, relying on a rule-based dependency parser for English, a hand-crafted translation lexicon (based on dictionary data extracted from parallel corpora by word alignment), various types of transfer rules, and a morphological generator for Finnish.

HY-NMT (**Raganato et al., 2018**) submissions are based on the Transformer “base" model, trained with all the parallel data provided by the shared task plus back-translations, with a shared vocabulary between source and target language and a domain label for each source sentence. For the multilingual sub-track synthetic data for English→Estonian and Estonian→English was also used. Ultimately, a single model for all language pairs was trained and then fine-tuned for each language pair.

HY-NMT-2STEP (**Raganato et al., 2018**) is a Transformer model trained on interleaved lemmas and morphological tags on the Finnish side. Morphological categories (number, tense etc.) have separate tags, and a tag is only added if the value of the category differs from the default value (in the same way that languages have morphemes only for marked values of morphological categories). The final translation is deterministically generated from the sequence of lemmas and morphological tags which the model outputs.

HY-SMT (**Tiedemann et al., 2016**) is the Helsinki SMT system submitted at WMT 2016 (the constrained-basic+back-translated version). The system was not retrained and it may thus suffer from poor lexical coverage on recent test data. The main motivation for including this baseline was to have a statistical machine translation (SMT) submission for the Finnish morphology test suite (Burlot et al., 2018).

### 2.3.9 JHU (Koehn et al., 2018a)

The JHU systems are the result of two relatively independent efforts on German–English language directions and Russian–English, using the Marian and Sockeye (Hieber et al., 2017) neural machine translation toolkits, respectively. The novel contributions are iterative back-translation (for German) and fine-tuning on test sets from prior years (for both languages).

### 2.3.10 JUCBNMT (Mahata et al., 2018)

JUCBNMT is an encoder-decoder sequence-to-sequence NMT model with character level encoding. The submission uses preprocessing like tokenization, truecasing and corpus cleaning. Both encoder and decoder use a single LSTM layer each. The batch size was set to 128, number of epochs was set to 100, activation function was softmax, optimizer chosen was RMSprop and the loss function used was categorical cross-entropy. Learning rate was set to 0.001.

### 2.3.11 KIT (Pham et al., 2018)

The KIT submission is the NMT Transformer architecture, enhanced in model depth. Techniques for reducing memory consumption (recalculating intermediate results at layers instead of caching them), 4 times larger model could fit on one GPU and improve the performance by 1.2 BLEU points.

Sentences selection from the new ParaCrawl improved the effectiveness of the corpus by 0.5 BLEU points, with an overall increase of 0.8 BLEU compared to the baseline of not using ParaCrawl.

### 2.3.12 LI-MUZE

LI-MUZE is an ensembles of 4 averaged Transformer models with one right-to-left and one target-to-source averaged Transformer model, the configuration of all the models is the same as the Transformer big-model, trained on the official training data with 4.5M back-translated data from the monolingual news of 2016 and 2017 data. The English vocabulary size is 36K BPE subwords. Chinese is tokenized by Chinese characters and the vocabulary size is 10K.

### 2.3.13 LMU-NMT (Huck et al., 2018)

For the WMT18 news translation shared task, LMU Munich (Huck et al., 2018) has trained basic shallow attentional encoder-decoder systems (Bahdanau et al., 2014) with the Nematus toolkit (Sennrich et al., 2017), like last year (Huck et al., 2017a). LMU has participated with these NMT systems for the English–German language pair in both translation directions. The training data is a concatenation of Europarl, News Commentary, Common Crawl, and some synthetic data in the form of backtranslated monolingual news texts. The 2017 monolingual News Crawl is not employed, nor are the parallel Rapid and ParaCrawl corpora. The German data is preprocessed with a linguistically informed word segmentation technique (Huck et al., 2017b). By using a linguistically more sound word segmentation, advantages over plain BPE segmentation are expected in three important aspects: vocabulary reduction, reduction of data sparsity, and open vocabulary translation. The NMT system can learn linguistic word formation processes from the segmented data. In the English→German translation direction, LMU furthermore conducted fine-tuning towards the domain of news articles (Huck et al., 2017a) and reranked the $n$-best list with a right-to-left neural model (Liu et al., 2016) which is trained for reverse word order (Freitag et al., 2013).

#### 2.3.14 LMU-UNSUP (Stojanovski et al., 2018)

For the unsupervised track of the WMT18 news translation task, LMU Munich submitted the LMU-UNSUP system (Stojanovski et al., 2018) which is a neural translation model trained without any access to parallel data. The model is trained with ∼4M German and English sentences each, which are sampled from NewsCrawl articles from 2007 to 2017. Bilingual word embeddings trained in an unsupervised manner (Conneau et al., 2017) were used to translate the monolingual data by doing word-by-word translation and this synthetically created parallel data is used in the training as well. The same model is used to do both German→English and English→German translation. The model is based on (Lample et al., 2018) and it uses denoising and on-the-fly backtranslation. Additionally the model uses the word-by-word translated data in the initial training stages to jump-start the training and disables the denoising component as the last training step for further improvements. The NMT embeddings are initialized with embeddings obtained from `fasttext` trained jointly on German and English monolingual BPE-level data.

#### 2.3.15 MICROSOFT-MARIAN (Junczys-Dowmunt, 2018)

MICROSOFT-MARIAN is the Transformer-big model implemented in Marian with an updated version of Edinburgh's training scheme for WMT2017, following current common practices: truecasing and tokenization using Moses scripts, BPE subwords, backtranslation (using a shallow model), ensembling of four left-to-right deep models and reranking of 12-best list with an ensemble of four right-to-left models.

The novelties are primarily in new data filtering (dual conditional cross-entropy filtering) and sentence weighting methods.

#### 2.3.16 MLLP-UPV (Iranzo-Sánchez et al., 2018)

MLLP-UPV is an ensemble of Transformer architecture-based neural machine translation systems. To train the system under “constrained" conditions, the provided parallel data was filtered with a scoring technique using character-based language models, and was augmented based on synthetic source sentences generated from the provided monolingual corpora.

The ensemble consists of 4 independent training runs of the Transformer “base” model, trained with 10M filtered sentences (including from ParaCrawl) and 20M backtranslated sentences from NewsCrawl2017.

#### 2.3.17 MMT-PRODUCTION

MMT-PRODUCTION is the machine translation system offered by MMT s.r.l. (`www.modernmt.eu`) as of July 2018. It is a Transformer-based neural MT system trained on public and proprietary data, containing about 100M sentence pairs and about 1.5G English words. It exploits a single model of ‘transformer-big’ size, and a single pass-decoding; texts are processed using internal tools.

#### 2.3.18 NEUROTOLGE.EE (Tars and Fishel, 2018)

NEUROTOLGE.EE is a multi-domain NMT system that treats text domain as language and applies the zero-shot multi-lingual approach to multiple domains in the training corpus. For WMT18, text domains were replaced with unsupervised clustering into 16 clusters using FastText's sentence embeddings. During translation the input segment is classified using its sentence embedding and translated as the corresponding cluster/domain.

#### 2.3.19 NICT (Marie et al., 2018)

NICT NMT systems were trained with the Transformer architecture using the provided parallel data enlarged with a large quantity of back-translated monolingual data generated with a new incremental training framework. The primary submissions to the task are the result of a simple combination between NICT SMT and NMT systems.

#### 2.3.20 NIUTRANS (Wang et al., 2018b)

NIUTRANS baseline systems are based on the Transformer architecture with the "base" model, equipped with checkpoint averaging and back-translation techniques. NIUTRANS further improve the translation performance 2.28-3.83 BLEU points from four aspects including model variances (larger inner-hidden-size in FFN, using ReLU and attention dropout, Swish activation function, relative positional representation), diverse ensemble decoding (ensemble decoding with up to 15 models, generated by different strategies), reranking (up to 14 features for reranking), and post-processing (aim at the inconsistent translation of proper nouns, especially the English literals in Chinese sentences).

#### 2.3.21 NJUNMT

The NJUNMT-PRIVATE is most likely the system developed by Natural Language Processing Group of Nanjing University based on high-level API of TensorFlow, https://github.com/zhaocq-nlp/NJUNMT-tf. Further details on training are not available.

#### 2.3.22 NTT (Morishita et al., 2018)

NTT combine Transformer "big" model, corpus cleaning technique for provided and synthetic parallel corpora, and right-to-left n-best re-ranking techniques. Through their experiments, NTT found filtering of noisy training sentences and right-to-left re-ranking as the keys to better accuracy.

#### 2.3.23 PARFDA (Biçici, 2018)

PARFDA selects a subset of the training and LM data to build task-specific SMT models. PARFDA uses phrase-based Moses and all constrained available resources provided by WMT18. The datasets are available at https://github.com/bicici/parfdaWMT2018.

#### 2.3.24 PROMT (Molchanov, 2018)

PROMT submitted three systems: PROMT-HYB-MARIAN, PROMT-HYB-OPENNMT and PROMT-RULE-BASED.

PROMT-HYB-MARIAN is an ensemble of 5 transformer models trained on WMT data and in-house news data.

PROMT-HYB-OPENNMT is a hybrid system based on PROMT Rule-based engine and a NMT post-editing (PE) engine. The NMT PE component is a sequence-to-sequence model with attention and deep biRNN encoder trained with OpenNMT toolkit.

PROMT-RULE-BASED is a rule-based system, without any specific training or tuning.

#### 2.3.25 RWTH (Schamper et al., 2018)

All systems submitted by RWTH Aachen for German to English are based on the Transformer architecture implemented in Sockeye. The final RWTH system has been an ensemble of three Transformer models, where each individual model had been already very strong. The strength of the RWTH systems is probably due to the following four key factors: (a) Using the Transformer architecture. (b) Rather large models and large batch size which was made possible due to synchronous training on 4 GPUs and roughly 8 days of training. (Details: num-embed: 1024; num-layers: 6; attention-heads: 16; transformer-feed-forward-num-hidden: 4096; transformer-model-size: 1024, no weight-tying. In sum, this results in 291M trainable parameters.) (c) Careful experiments on data conditions: E.g. oversampling of parallel data, LM driven filtering of ParaCrawl (retained 50%), testing different amounts of BPE merge operations. (d) Fine-tuning on old testsets (newstest2008-newstest2014).

RWTH English→Turkish system is based on 6-layer encoder-decoder Transformer architecture. Since the task has low resources, dropout with the rate of 0.3 to all applicable layers was used. Even though the two languages are not much related, joint BPE and weight tying helped a lot as part of regularization. For the final submission, RWTH used augmented training data with 1M-sentence back-translations and ensembled four models with different random seeds.

#### 2.3.26 RWTH-UNSUPER (Graça et al., 2018)

The RWTH-UNSUPER unsupervised NMT system is built based on recent works by Lample et al. (2018) and Artetxe et al. (2018). RWTH-UNSUPER best performing systems follow the batch optimization strategy and are initialized with cross-lingual embeddings. Furthermore, RWTH-UNSUPER found that sharing a vocabulary performs better than having separate ones. Freezing embeddings hurts performance and it was found best to initialize embeddings with pre-trained ones and train them as usual.

#### 2.3.27 TALP-UPC (Casas et al., 2018)

TALP-UPC is the Transformer "base" model trained with the Tensor2Tensor implementation (Vaswani et al., 2018) and wordpieces vocabulary. The training corpus is multilingual (concatenating Finnish–English and Estonian–English) and includes ParaCrawl with garbage cleaned up via `langdetect`.

#### 2.3.28 TENCENT (Wang et al., 2018a)

TENCENT-ENSEMBLE (called TenTrans) is an improved NMT system on Transformer based on self-attention mechanism. In addition to the basic settings of Transformer training, TENCENT-ENSEMBLE uses multi-model fusion techniques, multiple features reranking, different segmentation models and joint learning. Additionally, data selection strategies were adopted to fine-tune the trained system, achieving a stable performance improvement.

An additional system paper (Hu et al., 2018) describes a non-primary submission.

#### 2.3.29 TILDE (Pinnis et al., 2018)

TILDE submitted four systems: TILDE-C-NMT, TILDE-C-NMT-COMB, TILDE-C-NMT-2BT and TILDE-NC-NMT.

TILDE-C-NMT are constrained English-Estonian and Estonian-English NMT systems that were deployed as ensembles of averaged factored data Transformer models. The models were trained using filtered parallel data and back-translated data in a 1-to-1 proportion. The parallel data were supplemented with synthetic data (generated from the same parallel data) that contain unknown token identifiers in order to acquire models that are more robust to unknown phenomena.

TILDE-C-NMT-COMB is a constrained Estonian-English NMT system that is a system combination of multiple constrained factored data NMT systems.

TILDE-C-NMT-2BT systems were trained using Sockeye and Transformer models. Before training the initial systems, parallel data were cleaned using the `parallel-corpora-tools`. Before back-translation, monolingual data were also filtered. After back-translation, the resulting synthetic corpora were filtered again. Intermediate systems were trained with the first batch of parallel+synthetic data. The back-translation and filtering process was performed a second time with additional monolingual data to train the final systems with parallel and two sets of synthetic data.

TILDE-NC-NMT are unconstrained English→Estonian and Estonian→English NMT systems that were deployed as averaged Transformer models. These models were also trained using back-translated data similarly to the constrained systems, however, the data, taking into account its relatively large size, was not factored.

#### 2.3.30 UBIQUS

The UBIQUS-NMT system is probably developed by the Ubiqus company (`www.ubiqus.com`). No further information is available.

#### 2.3.31 UCAM (Stahlberg et al., 2018)

UCAM is a generalization of previous work (de Gispert et al., 2017) to multiple architectures. It is a system combination of two Transformer-like models, a recurrent model, a convolutional model, and a phrase-based SMT system. The output is probably dominated by the Transformer, and to some extend by the SMT system.

#### 2.3.32 UEDIN (Haddow et al., 2018)

For Estonian↔English and Finnish↔English, the UEDIN systems are an ensemble of four left-to-right systems, reranked with four right-to-left systems, built using Marian. Each ensemble consists of two Transformers and two deep RNNs. The RNNs use the UEDIN multi-head / multi-hop variant. All available parallel data were used, plus back-translated data from 2017 (for into-English) and 2014-2017 (for out-of-English). The natural parallel data was generally over-sampled to give an equal mix of parallel and synthetic data.

For English↔Estonian, UEDIN selected 30% of ParaCrawl based on translation model perplexity for a model built on the rest of the data.

The UEDIN systems for other language pairs use an ensemble of four deep RNN left-to-right systems, reranked with 4 deep RNN right-to-left systems. The RNN models use the UEDIN multi-head / multi-hop attention variant. All the provided parallel data (including ParaCrawl) were used, applying `langid` filtering to remove some incorrect sentence pairs. Synthetic data were also used, created by back-translating the 2017 English news crawl, and the 2017 and 2016 Czech news crawls. For Czech→English, the synthetic data was oversampled 2x.

#### 2.3.33 UMD (Xu and Carpuat, 2018)

The UMD best system is an ensemble of three 6-layer left-to-right Transformer models reranked with target-to-source and left-to-right models. Each Transformer model is trained with a 2:1 mixture of parallel and backtranslated monolingual data. For parallel data, duplicates are removed and "bad" sentence pairs filtered out. Monolingual data is sub-sampled from news 2017 (English) and news 2011 (Chinese). Subwords (BPE) are used for both English and Chinese sentences.

#### 2.3.34 UNISOUND

The UNISOUND systems are probably developed by the Unisound company (`www.unisound.com`). No further information is available.

#### 2.3.35 UNSUPTARTU (Del et al., 2018)

UNSUPTARTU is an unsupervised MT system using $n$-gram embedding cross-lingual mapping to create a phrase table. An RNN LM is used in decoding.

### 2.4 Multilingual Sub-track

This year the news translation track included an explicit sub-track on multilingual translation. This covered any submissions that used any data (monolingual or parallel) from a third language to help the language pair in question: for example, using English-Finnish data to improve English-Estonian translation. All entries to this sub-track had to use only the WMT-provided data sets, and thus had to be constrained. Submissions to this sub-track are joined with the main translation track and evaluated without separation in the same way.

While there was no restriction in terms of language pairs, three language pairs were "verbally endorsed": English to/from Turkish, Estonian and German. The motivation behind the choice of languages was to test the effect of multilingual (and unsupervised) methods on low-resource language pairs (Turkish-English, Estonian-English) and to contrast the results with a resource-rich pair (German-English).

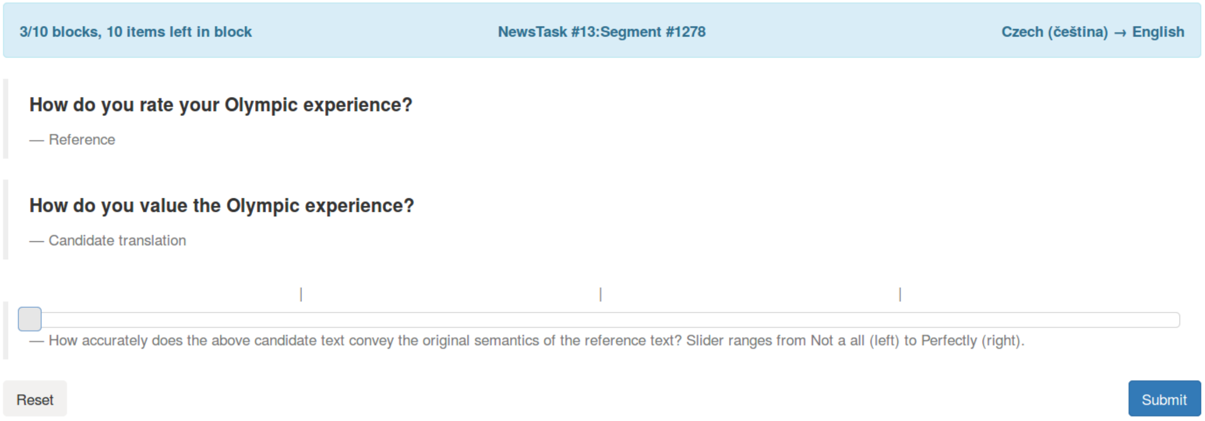
### 2.5 Unsupervised Sub-track

In the unsupervised MT sub-track the participants were constrained to using only the monolingual training data from WMT; this additionally excluded the monolingual corpora that are largely parallel (monolingual parts of Europarl and News Commentary). The aim of this task was to see how far can one get in terms of translation quality without any parallel data used for training. As an exception it was allowed to use a parallel dev set for parameter tuning and/or model selection. The language pairs of this sub-track coincided with the multilingual sub-track: English to/from Turkish, Estonian and German.

## 3 Human Evaluation

A human evaluation campaign is run each year to assess translation quality and to determine the final ranking of systems taking part in the competition. This section describes how preparation of evaluation data, collection of human assessments, and computation of the official results of the shared task was carried out this year.

Work on evaluation over the past few years has provided fresh insight into ways to collect *direct assessments* (DA) of machine translation quality (Graham et al., 2013, 2014, 2016), and two years ago the evaluation campaign included parallel assessment of a subset of News task language pairs evaluated with *relative ranking* (RR) and DA. DA has some clear advantages over RR, namely the evaluation of absolute translation quality and the ability to carry out evaluations through quality controlled crowd-sourcing. As established in 2016 (Bojar et al., 2016a), DA results (via crowd-sourcing) and RR results (produced by researchers) correlate strongly, with Pearson correlation ranging from 0.920 to 0.997 across several source languages into English and at 0.975 for English-to-Russian (the only pair evaluated out-of-English). Last year, we thus employed DA for evaluation of systems taking part in the news task and do so again this year. Where possible,


3/10 blocks, 10 items left in block
NewsTask #13:Segment #1278
Czech (čeština) → English

How do you rate your Olympic experience?

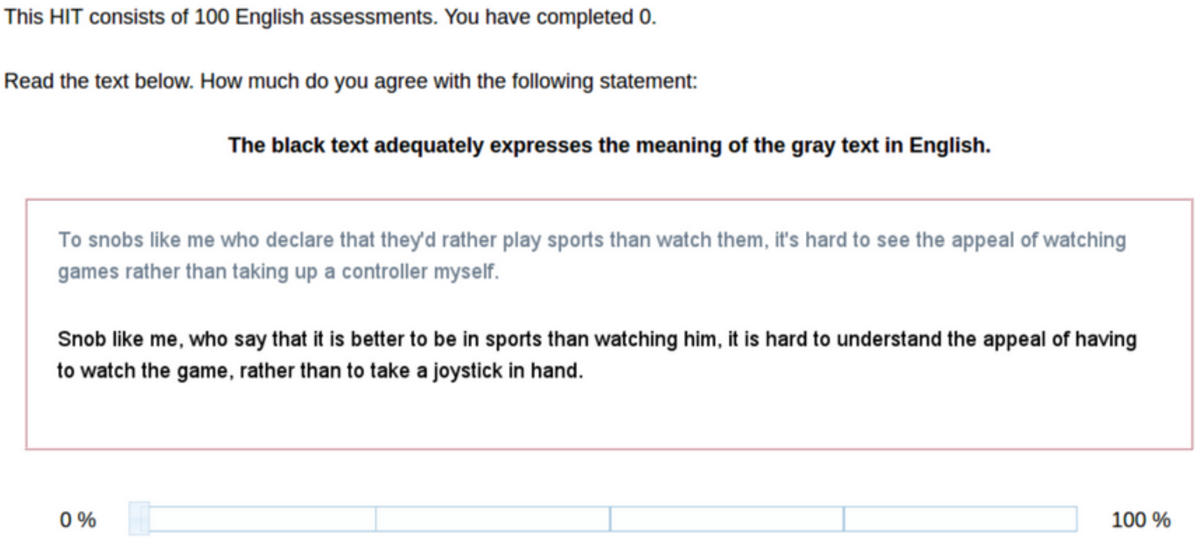
— Reference

How do you value the Olympic experience?

— Candidate translation

— How accurately does the above candidate text convey the original semantics of the reference text? Slider ranges from Not a all (left) to Perfectly (right).

**Figure 3:** Screen shot of Direct Assessment in the Appraise interface used in the human evaluation campaign. The annotator is presented with a reference translation and a single system output randomly selected from competing systems (anonymized), and is asked to rate the translation on a sliding scale.


This HIT consists of 100 English assessments. You have completed 0.

Read the text below. How much do you agree with the following statement:

**The black text adequately expresses the meaning of the gray text in English.**

To snobs like me who declare that they'd rather play sports than watch them, it's hard to see the appeal of watching games rather than taking up a controller myself.

Snob like me, who say that it is better to be in sports than watching him, it is hard to understand the appeal of having to watch the game, rather than to take a joystick in hand.

**Figure 4:** Screen shot of Direct Assessment as carried out by workers on Mechanical Turk.

we collect DA judgments via the crowd-sourcing platform, Amazon's Mechanical Turk, and as in previous year's we ask participating teams to provide manual evaluation of system outputs via Appraise. Researcher involvement was needed particularly for translations into Czech, German, Estonian, Finnish and Turkish.

Human assessors are asked to rate a given translation by how adequately it expresses the meaning of the corresponding reference translation (i.e. no bilingual speakers are needed) on an analogue scale, which corresponds to an underlying absolute 0–100 rating scale. Since DA involves evaluation of a single translation per screen, this allows the sentence length restriction usually applied during manual evaluation to be removed for both researchers and crowd-sourced workers.[6] Figure 3 shows one DA screen as completed by researchers on Appraise, while Figure 4 provides a screenshot of DA shown to crowd-sourced workers on Amazon's Mechanical Turk.

The annotation is organized into "HITs" (following the Mechanical Turk's term "human intelligence task"), each containing 100 such screens and requiring about half an hour to finish. Appraise users were allowed to pause their annotation at any time, Amazon interface did not allow any pauses. More details of composition of HITs are given in Section 3.3 below.

### 3.1 Evaluation Campaign Overview

In terms of the News translation task manual evaluation, a total of 584 individual researcher ac-

[6] The maximum sentence length with RR was 30 in WMT16.

counts were involved, and 915 turker accounts.[7] Researchers in the manual evaluation came from 33 different research groups and contributed judgments of 118,705 translations, while 225,900 translation assessment scores were submitted in total by the crowd.[8]

Under ordinary circumstances, each assessed translation would correspond to a single individual scored segment. However, since distinct systems can produce the same output for a particular input sentence, we are often able to take advantage of this and use a single assessment for multiple systems. Similar to last year's evaluation, we only combine human assessments in this way if the string of text belonging to multiple systems is exactly identical. For example, even small differences in punctuation disqualify combination of similar system outputs, and this is due to a general lack of evidence about what kinds of minor differences may or may not impact human evaluation.

Table 3 shows the numbers of segments for which distinct MT systems participating in the News Translation Task produced identical outputs. The biggest saving in terms of exact duplicate translations, being produced by multiple systems, was for German to English, where a 17.4% saving of resources by combining identical outputs before human evaluation.

### 3.2 Data Collection

System rankings are produced from a large set of human assessments of translations, each of which indicates the absolute quality of the output of a system. Annotations are collected in an evaluation campaign that enlists the help of participants in the shared task. Each team is asked to contribute 8 hours annotation time, which we estimated at 16 100-translation HITs per primary system submitted. We continue to use the open-source Appraise[9] (Federmann, 2012) tool for our data collection, in addition to Amazon Mechanical Turk.[10] Table 4 shows total numbers of human assessments collected in WMT18 contributing to final scores for systems.[11]

The effort that goes into the manual evaluation campaign each year is impressive, and we are grateful to all participating individuals and teams. We believe that human annotation provides the best decision basis for evaluation of machine translation output and it is great to see continued contributions on this large scale.

### 3.3 Crowd Quality Control

This year, two distinct HIT structures were run in the overall evaluation campaign, the standard DA set-up was employed for Mechanical Turk and a portion of the Appraise evaluation, while an additional HIT structure was used for the remaining part of the Appraise evaluation. Below we firstly describe the standard DA HIT structure and quality control mechanism before describing the additional version used for part of the Appraise evaluation. In both set-ups, translations are arranged in sets of 100-translation HITs to provide control over assignment and positioning of quality control items to human annotators.

In the standard DA HIT structure, three kinds of quality control translation pairs are employed as described in Table 5: we repeat pairs (expecting a similar judgment), damage MT outputs (expecting significantly worse scores) and use references instead of MT outputs (expecting high scores).

In total, 60 items in a 100-translation HIT serve in quality control checks but 40 of those are regular judgments of MT system outputs (we exclude assessments of bad references and ordinary reference translations when calculating final scores). The effort wasted for the sake of quality control is thus 20%.

Also in the standard DA HIT structure, within each 100-translation HIT, the same proportion of translations are included from each participating system for that language pair. This ensures the final dataset for a given language pair contains roughly equivalent numbers of assessments for each participating system. This serves three purposes for making the evaluation fair. Firstly, for the point estimates used to rank systems to be reliable, a sufficient sample size is needed and the

[7] Numbers do not include the 1,533 workers on Mechanical Turk and 7 Appraise evaluators who did not pass quality control.

[8] Numbers include quality control items for workers who passed quality control but omit the additional 347,700 assessments collected on Mechanical Turk where a worker did not pass quality control and equivalent 1,466 judgments for the small number of Appraise workers who did not meet the quality control threshold. A 40% pass rate for quality control is typical of DA evaluations on Mechanical Turk.

[9] https://github.com/cfedermann/Appraise

[10] https://www.mturk.com

[11] Appraise ran evaluation of $150 - 1 = 149$ systems due to a single tr-en system having been omitted in the initial human evaluation run. The 95 crowd-sourced systems includes all into-English language pair (including the tr-en missing system), en-ru and en-zh systems.

| Language Pair | Systems | Segments | Total Segments | Distinct Segments | Saving (%) WMT18 | Saving (%) WMT17 |
|---|---|---|---|---|---|---|
| Chinese→English | 14 | 3,981 | 55,734 | 49,767 | **10.7** | 3.9 |
| Czech→English | 5 | 2,983 | 14,915 | 13,987 | **6.2** | 4.3 |
| German→English | 16 | 2,998 | 47,968 | 39,627 | **17.4** | 10.7 |
| Estonian→English | 14 | 2,000 | 28,000 | 25,612 | **8.5** | – |
| Finnish→English | 9 | 3,000 | 27,000 | 25,233 | **6.5** | 1.4 |
| Russian→English | 8 | 3,000 | 24,000 | 21,966 | **8.5** | 5.8 |
| Turkish→English | 6 | 3,000 | 18,000 | 17,000 | **5.6** | 4.6 |
| English→Chinese | 14 | 3,981 | 55,734 | 48,022 | **13.8** | 1.7 |
| English→Czech | 5 | 2,983 | 14,915 | 13,982 | **6.3** | 10.2 |
| English→German | 16 | 2,998 | 47,968 | 39,963 | **16.7** | 12.8 |
| English→Estonian | 14 | 2,000 | 28,000 | 25,837 | **7.7** | – |
| English→Finnish | 12 | 3,000 | 36,000 | 32,749 | **9.0** | 3.7 |
| English→Russian | 9 | 3,000 | 27,000 | 24,594 | **8.9** | 4.5 |
| English→Turkish | 8 | 3,000 | 24,000 | 21,880 | **8.8** | 2.1 |

**Table 3:** Total segments prior to sampling for manual evaluation and savings made by combining outputs produced by different systems that were identical.

most efficient way to reach a sufficient sample size for all systems is to keep total numbers of judgments roughly equal as more and more judgments are collected. Secondly, it helps to make the evaluation fair because each system will suffer or benefit equally from an overly lenient/harsh human judge. Thirdly, despite DA judgments being absolute, it is known that judges "calibrate" the way they use the scale depending on the general observed translation quality. With each HIT including all participating systems, this effect is averaged out. Furthermore apart from quality control items, HITs are constructed using translations sampled from the entire set of outputs for a given language pair.

The alternate DA HIT structure employed by Appraise this year for a subset of researcher HITs is shown in Table 6. This set-up reduces the number of quality control items in a HIT and is therefore more efficient (12% overhead) by omitting repeat pairs and good reference pairs. This comes at the cost of a reduced ability to analyze the quality of data provided by human annotators. In addition for this set-up, an additional constraint (not originally applied in standard DA) was imposed. As much as possible within a 100-translation HIT the HIT included the output of all participating systems for each source input. This constraint has the advantage of producing assessments from the same human assessor for translations of the same source input but is not ideal in terms of the original aim of DA – to as much as possible produce absolute scores for translations (as opposed to relative ones) – because it positions assessment of competing translations in close proximity within a HIT and judges may attempt to remember their judgment for a different candidate translation of a given input sentence.

In all set-ups employed in the evaluation campaign, and as in previous years, bad reference pairs were created automatically by replacing a phrase within a given translation with a phrase of the same length randomly selected from n-grams extracted from the full test set of reference translations belonging to that language pair. This means that the replacement phrase will itself comprise a fluent sequence of words (making it difficult to tell that the sentence is low quality without reading the entire sentence) while at the same time making its presence highly likely to sufficiently change the meaning of the MT output so that it causes a noticeable degradation. The length of the phrase to be replaced is determined by the number of words in the original translation, as follows:

| Language Pair | Systems | Comps | Comps/Sys | Assessments | Assess/Sys |
|---|---|---|---|---|---|
| Chinese→English | 14 | – | – | 32,919 | 2,351.4 |
| Czech→English | 5 | – | – | 12,209 | 2,441.8 |
| German→English | 16 | – | – | 48,469 | 3,029.3 |
| Estonian→English | 14 | – | – | 28,868 | 2,062.0 |
| Finnish→English | 9 | – | – | 18,868 | 2,096.4 |
| Russian→English | 8 | – | – | 17,711 | 2,213.9 |
| Turkish→English | 6 | – | – | 29,784 | 4,964.0 |
| English→Chinese | 14 | – | – | 32,411 | 2,315.1 |
| English→Czech | 5 | – | – | 10,080 | 2,016.0 |
| English→German | 16 | – | – | 13,754 | 859.6 |
| English→Estonian | 14 | – | – | 15,800 | 1,128.6 |
| English→Finnish | 12 | – | – | 9,995 | 832.9 |
| English→Russian | 9 | – | – | 27,977 | 3,108.6 |
| English→Turkish | 8 | – | – | 3,644 | 455.5 |
| Total Researcher | 149 | – | – | 101,189 | 679.1 |
| Total Crowd | 95 | – | – | 201,300 | 2,118.9 |
| **Total WMT18** | **150** | – | – | **302,489** | **2,016.6** |
| WMT17 | 153 | – | – | 307,707 | 2,011.2 |
| WMT16 | 138 | 569,287 | 4,125.2 | 284,644 | 2,062.6 |
| WMT15 | 131 | 542,732 | 4,143.0 | 271,366 | 2,071.5 |
| WMT14 | 110 | 328,830 | 2,989.3 | 164,415 | 1,494.7 |
| WMT13 | 148 | 942,840 | 6,370.5 | 471,420 | 3,185.3 |
| WMT12 | 103 | 101,969 | 999.6 | 50,985 | 495.0 |
| WMT11 | 133 | 63,045 | 474.0 | 31,522 | 237.0 |

**Table 4:** Amount of data collected in the WMT18 manual evaluation campaign (assessments after removal of quality control items and "de-collapsing" *multi-system outputs*). The final seven rows report summary information from previous years of the workshop.

| Translation Length (N) | # Words Replaced in Translation |
|---|---|
| 1 | 1 |
| 2–5 | 2 |
| 6–8 | 3 |
| 9–15 | 4 |
| 16–20 | 5 |
| >20 | $\lfloor$ N/4 $\rfloor$ |

### 3.4 Annotator Agreement

When an analogue scale (or 0–100 point scale, in practice) is employed, agreement cannot be measured using the conventional Kappa coefficient, ordinarily applied to human assessment when judgments are discrete categories or preferences. Instead, to measure consistency we filter crowd-sourced human assessors by how consistently they rate translations of known distinct quality using the bad reference pairs described previously. Quality filtering via bad reference pairs is especially important for the crowd-sourced portion of the manual evaluation. Due to the anonymous nature of crowd-sourcing, when collecting assessments of translations, it is likely to encounter workers who attempt to game the service, as well as submission of inconsistent evaluations and even robotic ones. We therefore employ DA's quality control mechanism to filter out low quality data, facilitated by the use of DA's analogue rating scale.

Assessments belonging to a given crowd-sourced worker who has not demonstrated that he/she can reliably score bad reference translations significantly lower than corresponding genuine system output translations are filtered out. A paired significance test is applied to test if degraded translations are consistently scored lower

| | | |
|---|---|---|
| **Repeat Pairs**: | Original System output (10) | An exact repeat of it (10); |
| **Bad Reference Pairs**: | Original System output (10) | A degraded version of it (10); |
| **Good Reference Pairs**: | Original System output (10) | Its corresponding reference translation (10). |

**Table 5:** Standard DA HIT structure quality control translation pairs hidden within 100-translation HITs, numbers of items are provided in parentheses.

| | | |
|---|---|---|
| **Bad Reference Pairs**: | Original System output (12) | A degraded version of it (12). |

**Table 6:** Additional DA HIT structure used for a portion of researchers in Appraise data collection, where quality control translation pairs hidden within 100-translation HITs, numbers of items are provided in parentheses in adapted version of DA used for a subset of researchers HITs.

than their original counterparts and the p-value produced by this test is used as an estimate of human assessor reliability. Assessments of workers whose p-value does not fall below the conventional 0.05 threshold are omitted from the evaluation of systems, since they do not reliably score degraded translations lower than corresponding MT output translations.

This year's assessment includes the first large-scale DA evaluation where quality control items were applied to assessments of a known-reliable group, comprised of the portion of researchers who completed HITs on Appraise with the original DA HIT structure. Although this group should be considered highly reliable compared to Mechanical Turk for example, we must however keep in mind that a small part of this group are in fact hired to complete assessments and their reliability could vary more than what would be expected of volunteer researchers.

Table 7 shows the number of workers in the crowd-sourced and researcher groups who met our filtering requirement by showing a significantly lower score for bad reference items compared to corresponding MT outputs, and the proportion of those who simultaneously showed no significant difference in scores they gave to pairs of identical translations.

A main observation to be taken from Table 7 is the difference in proportions of human assessors on Mechanical Turk versus researchers who passed the quality filtering criteria for DA, by scoring degraded translations significantly lower than the original MT output counterparts, as 37% of Mechanical Turk workers were deemed reliable compared to 93% of evaluators in the researcher group. This low rate of workers passing quality filtering is inline with past DA evaluations, and the high proportion of annotators passing quality control is expected of a mostly known-reliable group. For crowd-sourced workers, consistent with past DA evaluations, Table 7 shows a substantially higher number of low quality workers encountered for evaluation of languages other than English on Mechanical Turk. For example, in the case of Russian and Chinese only a respective 22% and 10% of workers were considered reliable enough to include their assessments in the evaluation, compared to around 42% on average for English evaluations.

When we examine repeat assessments of the same translation, both filtered groups show similar levels of reliability with 96% of filtered Mechanical Turk workers and 95% of researchers showing no significant difference in scores for repeat assessment of the same translation. The idea is that the repeated input should receive a very similar score. Assuming that annotators do not remember their previous assessment for the repeated sentence, the "Exact Rep." corresponds to intra-annotator agreement and it reaches very high scores.[12]

Within the researcher group, although assessors have high levels of reliability overall, reliability in this respect varies quite a bit for different languages. For example, only 75% of assessors in the researcher group completing assessments for Estonian showed no significant difference for repeat assessment of the same translation, and 87% for Turkish, both lower levels of reliability than usually encountered on Mechanical Turk even though the research group is expected to be more reliable than crowd-sourced workers. However, on closer inspection, the number of human assessors who took part in the Turkish and Estonian evaluations is small and the seemingly large difference in percentages in fact correspond to as few as three indi-

[12] Repeat items are separated by a minimum of 40 intervening assessments to reduce the likelihood of annotators simply remembering previous scores for repeat assessment of translations.

|  |  |  | (A) | (B) |
|---|---|---|---|---|
|  |  |  | Sig. Diff. | (A) & No Sig. Diff. |
|  |  | All | Bad Ref. | Exact Rep. |
| **Mechanical Turk Crowd** | Czech→English | 169 | 74 ( 44%) | 70 ( 95%) |
|  | German→English | 514 | 227 ( 44%) | 216 ( 95%) |
|  | Estonian→English | 397 | 157 ( 40%) | 150 ( 96%) |
|  | Finnish→English | 238 | 102 ( 43%) | 99 ( 97%) |
|  | Russian→English | 203 | 96 ( 47%) | 93 ( 97%) |
|  | Turkish→English | 480 | 172 ( 36%) | 166 ( 97%) |
|  | Chinese→English | 401 | 153 ( 38%) | 148 ( 97%) |
|  | English→Russian | 209 | 47 ( 22%) | 45 ( 96%) |
|  | English→Chinese | 406 | 39 ( 10%) | 37 ( 95%) |
|  | **Crowd** | **2,477** | **915 ( 37%)** | **880 ( 96%)** |
| **Researcher** | German→English | 41 | 39 ( 95%) | 37 ( 95%) |
|  | Estonian→English | 16 | 13 ( 81%) | 13 (100%) |
|  | Finnish→English | 3 | 3 (100%) | 3 (100%) |
|  | Russian→English | 8 | 8 (100%) | 8 (100%) |
|  | Turkish→English | 7 | 7 (100%) | 7 (100%) |
|  | Chinese→English | 4 | 3 ( 75%) | 3 (100%) |
|  | English→Czech | 17 | 17 (100%) | 17 (100%) |
|  | English→German | 48 | 47 ( 98%) | 44 ( 94%) |
|  | English→Estonian | 6 | 4 ( 67%) | 3 ( 75%) |
|  | English→Finnish | 29 | 27 ( 93%) | 25 ( 93%) |
|  | English→Russian | 26 | 25 ( 96%) | 24 ( 96%) |
|  | English→Turkish | 17 | 15 ( 88%) | 13 ( 87%) |
|  | English→Chinese | 34 | 31 ( 91%) | 30 ( 97%) |
|  | **Researcher** | **256** | **239 ( 93%)** | **227 ( 95%)** |
| **Researcher**$_{alt}$ | Czech→English | 32 | 30 ( 94%) | — |
|  | German→English | 41 | 39 ( 95%) | — |
|  | Estonian→English | 12 | 12 (100%) | — |
|  | Finnish→English | 4 | 3 ( 75%) | — |
|  | Russian→English | 7 | 5 ( 71%) | — |
|  | Turkish→English | 3 | 2 ( 66%) | — |
|  | Chinese→English | 4 | 4 (100%) | — |
|  | English→Czech | 49 | 49 (100%) | — |
|  | English→German | 31 | 31 (100%) | — |
|  | English→Estonian | 83 | 83 (100%) | — |
|  | English→Finnish | 30 | 30 (100%) | — |
|  | English→Russian | 37 | 36 ( 97%) | — |
|  | English→Turkish | 6 | 6 (100%) | — |
|  | English→Chinese | 23 | 22 ( 96%) | — |
|  | **Researcher**$_{alt}$ | **362** | **352 ( 97%)** | — |
|  | **Total WMT18** | **3,095** | **1,506 ( 49%)** | **1,107 ( 96%)** |

**Table 7:** Number of unique workers, (A) those whose scores for bad reference items were significantly lower than corresponding MT outputs; (B) those of (A) whose scores also showed no significant difference for exact repeats of the same translation. *Researcher* denotes the portion of the evaluation carried out with the standard DA HIT structure, while *Researcher*$_{alt}$ denotes the remaining part that employed the altered HIT structure in which some quality control items are omitted.

viduals.

### 3.5 Producing the Human Ranking

All research and crowd data that passed quality control were combined to produce the overall shared task results. In order to iron out differences in scoring strategies of distinct human assessors, human assessment scores for translations were first standardized according to each individual human assessor's overall mean and standard deviation score, for both researchers and crowd. Average standardized scores for individual segments belonging to a given system are then computed, before the final overall DA score for that system is computed as the average of its segment scores (Ave $z$ in Table 8). Results are also reported for average scores for systems, computed in the same way but without any score standardization applied (Ave % in Table 8).

Table 8 includes final DA scores for all systems participating in WMT18 News Translation Task. Clusters are identified by grouping systems together according to which systems significantly outperform all others in lower ranking clusters, according to Wilcoxon rank-sum test.

Note that for English→German, the system FACEBOOK-FAIR is not considered a regular participant, but an invited/late submission, see Section 2.3.6.

Appendix A shows the underlying head-to-head significance test results for all pairs of systems.

### 3.6 Source-based Direct Assessment

A secondary bilingual manual evaluation was carried out involving an adaptation of the standard monolingual DA evaluation in which the source language input segment was used in place of the reference. Figure 5 provides a screenshot of this evaluation as implemented in Appraise, which we refer to as *source-based DA*. In this set-up system outputs are evaluated by bilinguals who have access to the source language input segment only and no reference translation. The main motivation for doing so was to free up reference translations to allow them to be used instead as a "human system" in the evaluation. By structuring the evaluation as a bilingual task it allows a human system to be manually evaluated under exactly the same conditions as all other systems thus providing an estimate of human performance.[13]

The aim of source-based DA is to produce accurate rankings for systems as well as the human system to allow direct comparison of system and human performance, motivated by recent indications that Machine Translation quality may in some cases be approaching human performance (Wu et al., 2016; Hassan et al., 2018). For source-based DA, annotators will ideally be bilingual, i.e. understand the source language sufficiently well, in addition to being native speakers of the target language. However, we did not specifically stipulate in this year's evaluation that human annotators be native speakers of the target language.

We run source-based DA for evaluation of English to Czech translation. This language pair was selected because sufficient annotators were available, helped by the fact that the set of systems participating in this language pair is small. This part of the campaign employs the alternate HIT structure described in Section 3.3 with reduced quality control items, i.e. it does not include exact repeats of translations or reference translations for quality control purposes.

A total of 17 annotators worked on the source-based DA pilot. 100% of annotators proved reliable, meaning that they scored bad reference items significantly lower than corresponding MT outputs (see Table 7 part (A) for corresponding reference-based DA percentages). For six candidate systems we collected $2,574$ assessments, resulting in an average of 429 annotations per individual system. Enforcing segment overlap during HIT creation resulted in 423 segments for which all six candidate translations have been scored. In total, annotators worked on 438 distinct segments.

Table 9 provides source-based DA scores for all primary English→Czech systems participating in WMT18 News Translation Task as well as the human system comprised of reference translations. Clusters are identified by grouping systems together according to which systems significantly outperform all others in lower ranking clusters, according to Wilcoxon rank-sum test.

As can be seen from clusters in Table 9, one system, CUNI-TRANSFORMER, appears to achieve quality better than that of the human reference, NEWSTEST2018-REF, while another, UEDIN, appears to be on par with human performance, and although both systems certainly achieve very impressive results, claims of *human parity* should be

[13] An alternate method is to keep DA monolingual but to employ secondary reference translations. No secondary reference translations were available for the test set, however.

**Chinese→English**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 78.8 | 0.140 | NiuTrans |
| | 77.7 | 0.111 | online-B |
| | 77.9 | 0.109 | UCAM |
| | 78.0 | 0.108 | Unisound-A |
| | 77.5 | 0.099 | Tencent-ensemble |
| | 77.5 | 0.094 | Unisound-B |
| | 77.9 | 0.091 | Li-Muze |
| | 77.0 | 0.089 | NICT |
| | 76.7 | 0.078 | UMD |
| 10 | 75.0 | −0.005 | online-Y |
| | 74.5 | −0.017 | uedin |
| 12 | 73.6 | −0.061 | online-A |
| 13 | 65.9 | −0.327 | online-G |
| 14 | 64.4 | −0.377 | online-F |

**English→Chinese**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 80.7 | 0.219 | Tencent-ensemble |
| | 80.3 | 0.206 | Unisound |
| | 80.5 | 0.199 | GTCOM-Primary |
| | 79.7 | 0.185 | Alibaba-Ens-Rerank |
| | 79.2 | 0.173 | Alibaba-General-A |
| | 79.5 | 0.166 | online-B |
| | 79.0 | 0.165 | Alibaba-General-B |
| 8 | 78.1 | 0.094 | UMD |
| | 77.5 | 0.082 | NICT |
| | 77.1 | 0.069 | online-Y |
| | 75.5 | 0.037 | online-A |
| 12 | 70.7 | −0.202 | uedin |
| 13 | 63.3 | −0.419 | online-F |
| | 63.4 | −0.435 | online-G |

**Czech→English**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 71.8 | 0.298 | CUNI-Transformer |
| 2 | 67.9 | 0.165 | uedin |
| 3 | 66.6 | 0.115 | online-B |
| 4 | 62.1 | −0.023 | online-A |
| 5 | 57.5 | −0.183 | online-G |

**English→Czech**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 67.2 | 0.594 | CUNI-Transformer |
| 2 | 60.6 | 0.384 | uedin |
| 3 | 52.1 | 0.101 | online-B |
| 4 | 46.0 | −0.115 | online-A |
| 5 | 42.0 | −0.246 | online-G |

**German→English**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 79.9 | 0.413 | RWTH |
| | 79.4 | 0.395 | UCAM |
| | 78.2 | 0.359 | NTT |
| | 77.3 | 0.346 | online-B |
| | 77.4 | 0.321 | MLLP-UPV |
| | 77.0 | 0.317 | JHU |
| | 76.9 | 0.315 | Ubiqus-NMT |
| | 76.7 | 0.310 | online-Y |
| | 75.7 | 0.268 | online-A |
| | 75.4 | 0.261 | uedin |
| 11 | 72.5 | 0.162 | LMU-nmt |
| | 72.2 | 0.149 | NJUNMT-private |
| 13 | 65.2 | −0.074 | online-G |
| 14 | 58.5 | −0.296 | online-F |
| 15 | 45.4 | −0.752 | RWTH-UNSUPER |
| 16 | 42.7 | −0.835 | LMU-unsup |

**English→German**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 85.5 | 0.653 | Facebook-FAIR ⋆ |
| 2 | 82.2 | 0.561 | online-B |
| | 81.9 | 0.551 | Microsoft-Marian |
| | 81.6 | 0.539 | MMT-production |
| | 82.3 | 0.537 | UCAM |
| | 80.2 | 0.491 | NTT |
| | 79.3 | 0.454 | KIT |
| 8 | 77.7 | 0.396 | online-Y |
| | 76.7 | 0.377 | JHU |
| | 76.3 | 0.352 | uedin |
| 11 | 71.8 | 0.213 | LMU-nmt |
| 12 | 67.4 | 0.060 | online-A |
| 13 | 53.2 | −0.385 | online-F |
| | 53.8 | −0.416 | online-G |
| 15 | 36.7 | −0.966 | RWTH-UNSUPER |
| 16 | 32.6 | −1.122 | LMU-unsup |

**Estonian→English**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 73.3 | 0.326 | tilde-nc-nmt |
| 2 | 71.1 | 0.238 | NICT |
| | 69.9 | 0.215 | tilde-c-nmt |
| | 69.0 | 0.187 | tilde-c-nmt-2bt |
| | 69.2 | 0.186 | uedin |
| | 68.7 | 0.171 | tilde-c-nmt-comb |
| | 67.1 | 0.117 | online-B |
| | 66.4 | 0.106 | HY-NMT |
| | 66.8 | 0.106 | talp-upc |
| 10 | 65.4 | 0.063 | online-A |
| | 64.0 | 0.007 | CUNI-Kocmi |
| 12 | 59.4 | −0.117 | neurotolge.ee |
| 13 | 52.7 | −0.341 | online-G |
| 14 | 34.6 | −0.950 | UnsupTartu |

**English→Estonian**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 64.9 | 0.549 | tilde-nc-nmt |
| 2 | 62.1 | 0.453 | NICT |
| | 61.6 | 0.427 | tilde-c-nmt |
| | 61.2 | 0.418 | tilde-c-nmt-2bt |
| 5 | 58.6 | 0.340 | Aalto |
| | 58.6 | 0.329 | HY-NMT |
| | 57.5 | 0.295 | uedin |
| 8 | 55.5 | 0.216 | CUNI-Kocmi |
| | 54.6 | 0.181 | talp-upc |
| 10 | 52.1 | 0.097 | online-B |
| 11 | 45.7 | −0.132 | neurotolge.ee |
| 12 | 43.8 | −0.195 | online-A |
| 13 | 37.6 | −0.406 | online-G |
| 14 | 34.3 | −0.520 | parfda |

**Finnish→English**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 75.2 | 0.153 | NICT |
| | 74.4 | 0.128 | HY-NMT |
| | 74.0 | 0.103 | uedin |
| | 72.7 | 0.083 | CUNI-Kocmi |
| | 72.9 | 0.078 | online-B |
| | 71.9 | 0.047 | talp-upc |
| | 71.5 | 0.045 | online-A |
| 8 | 66.1 | −0.134 | online-G |
| 9 | 58.9 | −0.404 | JUCBNMT |

**English→Finnish**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 64.7 | 0.521 | NICT |
| | 63.1 | 0.466 | HY-NMT |
| 3 | 59.2 | 0.324 | uedin |
| | 58.3 | 0.271 | Aalto |
| | 57.9 | 0.258 | HY-NMT-2step |
| | 57.4 | 0.238 | talp-upc |
| | 55.9 | 0.184 | CUNI-Kocmi |
| | 56.6 | 0.183 | online-B |
| 9 | 45.9 | −0.212 | online-A |
| | 45.3 | −0.233 | online-G |
| 11 | 42.7 | −0.334 | HY-SMT |
| | 41.5 | −0.369 | HY-AH |

**Russian→English**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 81.0 | 0.215 | Alibaba |
| | 80.3 | 0.192 | online-B |
| | 79.6 | 0.170 | online-G |
| 4 | 77.5 | 0.110 | uedin |
| 5 | 76.2 | 0.034 | online-A |
| 6 | 74.1 | −0.014 | afrl-syscomb |
| | 73.7 | −0.027 | JHU |
| 8 | 64.2 | −0.398 | online-F |

**English→Russian**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 72.0 | 0.352 | Alibaba-Ens |
| | 71.4 | 0.324 | online-G |
| 3 | 66.8 | 0.159 | online-B |
| | 66.0 | 0.144 | uedin |
| | 64.9 | 0.115 | PROMT-Hyb-Marian |
| 6 | 63.9 | 0.066 | PROMT-Hyb-OpenNMT |
| 7 | 62.2 | −0.004 | online-A |
| 8 | 59.1 | −0.075 | PROMT-Rule-based |
| 9 | 44.5 | −0.580 | online-F |

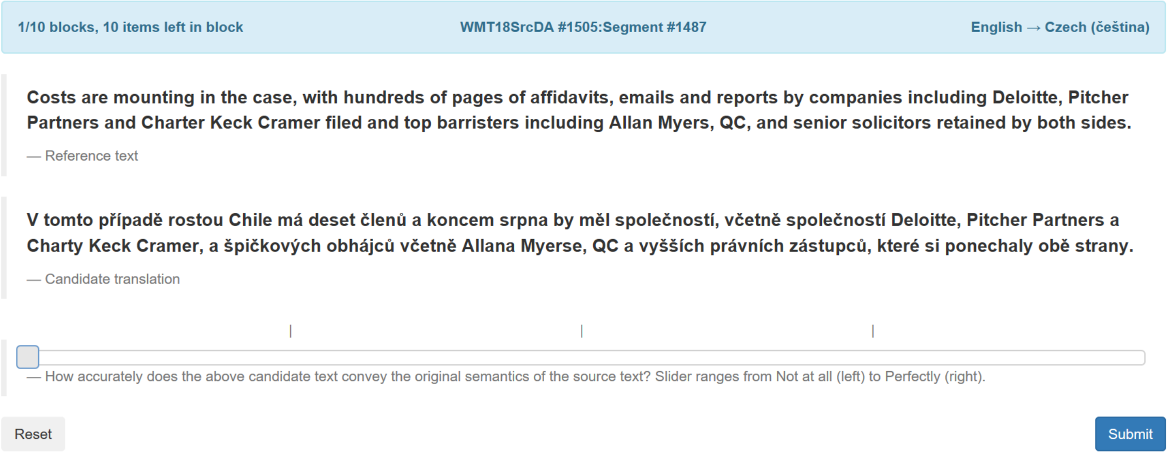
**Turkish→English**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 70.2 | 0.101 | online-G |
| | 69.3 | 0.077 | online-A |
| | 68.1 | 0.030 | Alibaba-Ens |
| | 68.0 | 0.027 | online-B |
| | 67.0 | −0.008 | uedin |
| | 66.0 | −0.040 | NICT |

**English→Turkish**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 66.3 | 0.277 | online-B |
| | 63.6 | 0.222 | uedin |
| | 63.5 | 0.216 | Alibaba-Ens-A |
| | 62.0 | 0.128 | NICT |
| | 60.1 | 0.111 | Alibaba-Ens-B |
| | 60.1 | 0.058 | online-G |
| 7 | 55.0 | −0.060 | RWTH |
| 8 | 49.6 | −0.254 | online-A |

**Table 8:** Official results of WMT18 News Translation Task. Systems ordered by standardized mean DA score, though systems within a cluster are considered tied. Lines between systems indicate clusters according to Wilcoxon rank-sum test at p-level $p < 0.05$. Systems with gray background indicate use of resources that fall outside the constraints provided for the shared task.



1/10 blocks, 10 items left in block

WMT18SrcDA #1505:Segment #1487

English → Czech (čeština)

Costs are mounting in the case, with hundreds of pages of affidavits, emails and reports by companies including Deloitte, Pitcher Partners and Charter Keck Cramer filed and top barristers including Allan Myers, QC, and senior solicitors retained by both sides.

— Reference text

V tomto případě rostou Chile má deset členů a koncem srpna by měl společností, včetně společností Deloitte, Pitcher Partners a Charty Keck Cramer, a špičkových obhájců včetně Allana Myerse, QC a vyšších právních zástupců, které si ponechaly obě strany.

— Candidate translation

— How accurately does the above candidate text convey the original semantics of the source text? Slider ranges from Not at all (left) to Perfectly (right).

**Figure 5:** Screen shot of source-based Direct Assessment in the Appraise interface used in the English→Czech pilot campaign. The annotator is presented with a source text and a single system output randomly selected from competing systems (anonymized), and is asked to rate the translation on a sliding scale.

taken with a degree of caution for several reasons which we outline below.

Firstly, the alternate HIT structure applied in this version of DA has not been tested thoroughly enough to be certain of high reliability. For example, as described in Section 3.3, forcing all translations of a given source segment to be assessed by the same human judge within the same HIT could cause individual DA ratings to become highly relative as opposed to the aim of DA ratings to be as close as possible to absolute judgments of translation quality. Furthermore, an additional bias that could cause problems for this HIT structure is one associated with a past evaluation method, relative ranking. When evaluating competing translations of the same source that are situated in close proximity within a HIT, annotators may be primed by high (or low) quality outputs resulting in overly severe (or lenient) judgments for subsequent translations of the same source segment (Bojar et al., 2011).

Secondly, while standard monolingual DA employs annotators only required to be speakers of a single language, source-based DA requires fluency in two languages and it is not known the degree to which varying levels of native language fluency in at least one language may negatively impact the reliability of DA rankings in the case of bilingual annotators.

Thirdly, it is likely that the quality of reference translations can vary and this could potentially impact the reliability of human performance estimates in source-based DA. Although reference-based DA assumes high quality reference translations, in the unfortunate case of problematic references, the overall rankings are unlikely to suffer to any large degree in terms of the reliability of system rankings, since all competing systems are likely to suffer equally from any lack of quality in reference translations.

However, in the adapted source-based version of DA, the effect of low quality reference translations is quite different. Firstly, since assessment involves comparison of MT outputs with the source, genuine participating systems will not suffer from the fact reference translations are low quality, since references are not involved in their evaluation. On the other hand, human performance estimates certainly will, as a drop in reference quality is indeed highly likely to negatively impact the placement of human performance estimates in system rankings. The reliability of comparisons with human performance with source-based DA is therefore highly dependent on high quality reference translations, as employment of a low quality set of references can only lead to *underestimates of human performance*. Considering the manual evaluation included several reports of ill-formed reference translations, conclusions of human parity and/or superiority relative to humans should be avoided.

### 3.7 Considerations as to Human Parity

As mentioned above, before making any statements about "machine translation outperforming humans" or "machine-human parity in translation" it may be important to consider the following ad-

**English→Czech**

| | Ave. % | Ave. z | System |
|---|---|---|---|
| 1 | 84.4 | 0.667 | CUNI-TRANSFORMER |
| 2 | 79.8 | 0.521 | UEDIN |
| | 78.6 | 0.483 | NEWSTEST2018-REF |
| 4 | 68.1 | 0.128 | ONLINE-B |
| 5 | 59.4 | −0.178 | ONLINE-A |
| 6 | 54.1 | −0.354 | ONLINE-G |

**Table 9:** Source-based DA results for English→Czech newstest2018, where systems are ordered by standardized mean DA score, though systems within a cluster are considered tied. Lines between systems indicate clusters according to Wilcoxon rank-sum test at p-level $p < 0.05$. Systems with gray background indicate use of resources that fall outside the constraints provided for the shared task. NEWSTEST2018-REF denotes the human system comprised of human-produced reference translations.

ditional points:

- Since none of WMT18 systems process larger units than individual sentences and our evaluation does not include any context beyond individual segments, it is possible that the human estimate is under-rewarded for correct cross-sentential phenomena.

- The sample size employed in the source-based DA evaluation was smaller than the recommended 1,500 judgments per system.

- The way in which translations in the test sets were originally created was as follows: one half of the test data for a given language pair was translated in one language direction and the other half in the opposite direction. It is well known that the translation direction affects translation quality in training and this could also be the case for evaluation.

- The formal education in linguistics or translatology of human assessors has not been taken into account: it is likely that whether or not human assessors have received any formal training in translation might influence their acceptance of varying levels of well-formedness in translations. For example, untrained assessors might not be as sensitive to subtle differences in verb conjugation, based on their own experience: In many real-life situations, the exact verb tense or conditional chosen in one sentence may not really impact the overall message because it can be implied from the context (and thus left free to the imagination of the annotator in our sentence-based evaluation) or from general knowledge.

## 4 Test Suites

Arguably, both the manual and automatic evaluations carried out at WMT News Translation Task are rather opaque. We learn (for each language pair and with a known confidence) which systems perform better *on average* over the sentences sampled from the news test set.

This average performance however does not provide any insight into *which particular phenomena* are handled better or worse by the systems. It is quite possible that the overall best-performing system may be unreliable for long sentences, for named entities, for pronouns or others. Such targeted evaluations may be important for particular deployment settings and use cases, and they are definitely important for us, MT system developers, in order to focus on them in subsequent research.

## Acknowledgments

This work was supported in part by funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nos. 645452 (QT21) and 645357 (Cracker), and from the Connecting Europe Facility under agreement No. NEA/CEF/ICT/A2016/1331648 (ParaCrawl).

We would also like to thank the University of Helsinki, the University of Tartu, Yandex and Microsoft for supplying test data for the news translation task.

The human evaluation campaign was very gratefully supported by contributions from Amazon, Microsoft, and Science Foundation Ireland in the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Dublin City University funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund.

We would also like to give special thanks to the small group of Turkish speakers who rescued our English-Turkish human evaluation at very short notice by contributing their time voluntarily. Finally, we are grateful to the large number of anonymous Mechanical Turk workers who contributed their human intelligence to the human evaluation.

## References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints*, abs/1409.0473. Presented at ICLR 2015.

Loïc Barrault, Fethi Bougares, Lucia Specia, Chiraag Lala, Desmond Elliott, and Stella Frank. 2018. Findings of the Third Shared Task on Multimodal Machine Translation. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Chao Bei, Hao Zong, Yiming Wang, Baoyong Fan, Shiqi Li, and Conghu Yuan. 2018. An Empirical Study of Machine Translation for the Shared Task of WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Ergun Biçici. 2018. Robust parfda Statistical Machine Translation Results. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Miloš Ercegovčević, Martin Popel, and Omar Zaidan. 2011. A Grain of Salt for the WMT Manual Evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, Edinburgh, Scotland. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016a. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016b. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Text, Speech and Dialogue: 19th International Conference, TSD 2016, Brno, Czech Republic, September 12-16, 2016, Proceedings*. Springer Verlag.

Ondřej Bojar, Jiří Mírovský, Kateřina Rysová, and Magdaléna Rysová. 2018. EvalD Reference-Less Discourse Evaluation for WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Franck Burlot, Yves Scherrer, Vinit Ravishankar, Ondřej Bojar, Stig-Arne Grönroos, Maarit Koponen, Tommi Nieminen, and François Yvon. 2018. The WMT'18 Morpheval test suites for English-Czech, English-German, English-Finnish and Turkish-English. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Chris Callison-Burch, Cameron Shaw Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (Meta-) Evaluation of Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic. Association for Computational Linguistics.

Chris Callison-Burch, Cameron Shaw Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further Meta-Evaluation of Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation. In *Proceedings of the Joint Fifth*

*Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–48, Montreal, Canada. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland. Association for Computational Linguistics.

Noe Casas, Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. 2018. The TALP-UPC Machine Translation Systems for WMT18 News Shared Translation Task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 Shared Task on Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word Translation Without Parallel Data. *CoRR*, abs/1710.04087.

Maksym Del, Andre Tättar, and Mark Fishel. 2018. Phrase-based Unsupervised Machine Translation with Compositional Phrase Embeddings. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, Changfeng Zhu, and Boxing Chen. 2018. Alibaba's Neural Machine Translation Systems for WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium. Association for Computational Linguistics.

Christian Federmann. 2012. Appraise: an open-source toolkit for manual evaluation of mt output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Markus Freitag, Minwei Feng, Matthias Huck, Stephan Peitz, and Hermann Ney. 2013. Reverse Word Order Models. In *Proceedings of the XIV Machine Translation Summit*, pages 159–166, Nice, France.

Markus Freitag, Matthias Huck, and Hermann Ney. 2014. Jane: Open Source Machine Translation System Combination. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32, Gothenburg, Sweden. Association for Computational Linguistics.

Adrià de Gispert, Bill Byrne, Eva Hasler, and Felix Stahlberg. 2017. Neural machine translation by minimising the bayes-risk with respect to syntactic translation lattices. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 362–368.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous Measurement Scales in Human Evaluation of Machine Translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2014. Is Machine Translation Getting Better over Time? In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 443–451, Gothenburg, Sweden. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, pages 1–28.

Miguel Graça, Yunsu Kim, Julian Schamper, Jiahui Geng, and Hermann Ney. 2018. The RWTH Aachen University English-German and German-English Unsupervised Neural Machine Translation Systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2018. Cognate-aware morphological segmentation for multilingual neural translation. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Jeremy Gwinnup, Tim Anderson, Grant Erdmann, and Katherine Young. 2018. The AFRL WMT18 Systems: Ensembling, Continuation and Combination. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Jeremy Gwinnup, Timothy Anderson, Grant Erdmann, Katherine Young, Michaeel Kazi, Elizabeth Salesky, Brian Thompson, and Jonathan Taylor. 2017. The AFRL-MITLL WMT17 Systems: Old, New, Borrowed, BLEU. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Barry Haddow, Nikolay Bogoychev, Denis Emelin, Ulrich Germann, Roman Grundkiewicz, Kenneth Heafield, Antonio Valerio Miceli Barone, and Rico Sennrich. 2018. The University of Edinburgh's Submissions to the WMT18 News Translation Task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving Human Parity on Automatic Chinese to English News Translation. https://www.microsoft.com/en-us/research/uploads/prod/2018/03/final-achieving-human.pdf.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1712.05690*.

Bojie Hu, Ambyer Han, and Shen Huang. 2018. TencentFmRD Neural Machine Translation for WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Matthias Huck, Fabienne Braune, and Alexander Fraser. 2017a. LMU Munich's Neural Machine Translation Systems for News Articles and Health Information Texts. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Matthias Huck, Simon Riess, and Alexander Fraser. 2017b. Target-side Word Segmentation Strategies for Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Matthias Huck, Dario Stojanovski, Viktor Hangya, and Alexander Fraser. 2018. LMU Munich's Neural Machine Translation Systems at WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Arvi Hurskainen and Jörg Tiedemann. 2017. Rule-based Machine translation from English to Finnish. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Javier Iranzo-Sánchez, Pau Baquero-Arnal, Gonçal V. Garcés Díaz-Munío, Adrià Martínez-Villaronga, Jorge Civera, and Alfons Juan. 2018. The MLLP-UPV German-English Machine Translation System for WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Microsoft's Submission to the WMT2018 News Translation Task: How I Learned to Stop Worrying and Love the Data. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in c++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Tom Kocmi, Roman Sudarikov, and Ondřej Bojar. 2018. CUNI Submissions in WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn, Kevin Duh, and Brian Thompson. 2018a. The JHU Machine Translation Systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher J. Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007.

Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018b. Findings of the WMT 2018 Shared Task on Parallel Corpus Filtering. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn and Christof Monz. 2006. Manual and Automatic Evaluation of Machine Translation between European Languages. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 102–121, New York City. Association for Computational Linguistics.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-Based & Neural Unsupervised Machine Translation. *arXiv preprint arXiv:1804.07755*.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on Target-bidirectional Neural Machine Translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416, San Diego, CA, USA. Association for Computational Linguistics.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay. 2018. JUCBNMT at WMT2018 News Translation Task: Character Based Neural Machine Translation of Finnish to English. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Benjamin Marie, Rui Wang, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. 2018. NICT's Neural and Statistical Machine Translation Systems for the WMT18 News Translation Task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Alexander Molchanov. 2018. PROMT Systems for WMT 2018 Shared Translation Task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2018. NTT's Neural Machine Translation Systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Mariana Neves, Antonio Jimeno Yepes, Aurélie Névéol, Cristian Grozea, Amy Siu, Madeleine Kittner, and Karin Verspoor. 2018. Findings of the WMT 2018 Biomedical Translation Shared Task: Evaluation on Medline test sets. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Ngoc-Quan Pham, Jan Niehues, and Alexander Waibel. 2018. The Karlsruhe Institute of Technology Systems for the News Translation Task in WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Mārcis Pinnis, Matīss Rikters, and Rihards Krišlauks. 2018. Tilde's Machine Translation Systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Martin Popel. 2018. CUNI Transformer Neural MT System for WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Martin Popel and Ondřej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Alessandro Raganato, Yves Scherrer, Tommi Nieminen, Arvi Hurskainen, and Jörg Tiedemann. 2018. The University of Helsinki submissions to the WMT18 news task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Julian Schamper, Jan Rosendahl, Parnia Bahar, Yunsu Kim, Arne Nix, and Hermann Ney. 2018. The RWTH Aachen University Supervised Machine Translation Systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón Astudillo, and André F. T. Martins. 2018. Findings of the WMT 2018 Shared Task on Quality Estimation. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2018. The University of Cambridge's Machine Translation Systems for WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Dario Stojanovski, Viktor Hangya, Matthias Huck, and Alexander Fraser. 2018. The LMU Munich Unsupervised Machine Translation Systems. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Roman Sudarikov, Martin Popel, Ondřej Bojar, Aljoscha Burchardt, and Ondřej Klejch. 2016. Using MT-ComparEval. In *Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem*, pages 76–82.

Sander Tars and Mark Fishel. 2018. Multi-Domain Neural Machine Translation. In *Proceedings of EAMT*, pages 259–268.

Jörg Tiedemann, Fabienne Cap, Jenna Kanerva, Filip Ginter, Sara Stymne, Robert Östling, and Marion Weller-Di Marco. 2016. Phrase-Based SMT for Finnish with More Data, Better Models and Alternative Alignment and Translation Tools. In *Proceedings of the First Conference on Machine Translation*, pages 391–398, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for Neural Machine Translation. *CoRR*, abs/1803.07416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python Implementation and Extensions for Morfessor Baseline. Technical report.

Mingxuan Wang, Li Gong, Wenhuan Zhu, Jun Xie, and Chao Bian. 2018a. Tencent Neural Machine Translation Systems for WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018b. The NiuTrans Machine Translation System for WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Weijia Xu and Marine Carpuat. 2018. The University of Maryland's Chinese-English Neural Machine Translation Systems at WMT18. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

## A Differences in Human Scores

Tables 10–23 show differences in average standardized human scores for all pairs of competing systems for each language pair. The numbers in each of the tables' cells indicate the difference in average standardized human scores for the system in that column and the system in that row.

Because there were so many systems and data conditions the significance of each pairwise comparison needs to be quantified. We applied Wilcoxon rank-sum test to measure the likelihood that such differences could occur simply by chance. In the following tables $\star$ indicates statistical significance at $p < 0.05$, † indicates statistical significance at $p < 0.01$, and ‡ indicates statistical significance at $p < 0.001$, according to Wilcoxon rank-sum test.

Each table contains final rows showing the average score achieved by that system and the rank range according according to Wilcoxon rank-sum test ($p < 0.05$). Gray lines separate clusters based on non-overlapping rank ranges.

Table 24 shows the differences in average standardized human scores for Czech→English systems, based on source-based DA.

| | NiuTrans | online-B | UCAM | Unisound-A | Tencent-ensemble | Unisound-B | Li-Muze | NICT | UMD | online-Y | uedin | online-A | online-G | online-F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NiuTrans | - | 0.03 | 0.03 | 0.03 | 0.04⋆ | 0.05⋆ | 0.05⋆ | 0.05⋆ | 0.06† | 0.15‡ | 0.16‡ | 0.20‡ | 0.47‡ | 0.52‡ |
| online-B | -0.03 | - | 0.00 | 0.00 | 0.01 | 0.02 | 0.02 | 0.02 | 0.03 | 0.12‡ | 0.13‡ | 0.17‡ | 0.44‡ | 0.49‡ |
| UCAM | -0.03 | 0.00 | - | 0.00 | 0.01 | 0.02 | 0.02 | 0.02 | 0.03 | 0.11‡ | 0.13‡ | 0.17‡ | 0.44‡ | 0.49‡ |
| Unisound-A | -0.03 | 0.00 | 0.00 | - | 0.01 | 0.01 | 0.02 | 0.02 | 0.03 | 0.11‡ | 0.12‡ | 0.17‡ | 0.43‡ | 0.48‡ |
| Tencent-ensemble | -0.04 | -0.01 | -0.01 | -0.01 | - | 0.01 | 0.01 | 0.01 | 0.02 | 0.10‡ | 0.12‡ | 0.16‡ | 0.43‡ | 0.48‡ |
| Unisound-B | -0.05 | -0.02 | -0.02 | -0.01 | -0.01 | - | 0.00 | 0.01 | 0.02 | 0.10‡ | 0.11‡ | 0.16‡ | 0.42‡ | 0.47‡ |
| Li-Muze | -0.05 | -0.02 | -0.02 | -0.02 | -0.01 | 0.00 | - | 0.00 | 0.01 | 0.10‡ | 0.11‡ | 0.15‡ | 0.42‡ | 0.47‡ |
| NICT | -0.05 | -0.02 | -0.02 | -0.02 | -0.01 | -0.01 | 0.00 | - | 0.01 | 0.09‡ | 0.11‡ | 0.15‡ | 0.42‡ | 0.47‡ |
| UMD | -0.06 | -0.03 | -0.03 | -0.03 | -0.02 | -0.02 | -0.01 | -0.01 | - | 0.08† | 0.10‡ | 0.14‡ | 0.40‡ | 0.45‡ |
| online-Y | -0.15 | -0.12 | -0.11 | -0.11 | -0.10 | -0.10 | -0.10 | -0.09 | -0.08 | - | 0.01 | 0.06⋆ | 0.32‡ | 0.37‡ |
| uedin | -0.16 | -0.13 | -0.13 | -0.12 | -0.12 | -0.11 | -0.11 | -0.11 | -0.10 | -0.01 | - | 0.04⋆ | 0.31‡ | 0.36‡ |
| online-A | -0.20 | -0.17 | -0.17 | -0.17 | -0.16 | -0.16 | -0.15 | -0.15 | -0.14 | -0.06 | -0.04 | - | 0.27‡ | 0.32‡ |
| online-G | -0.47 | -0.44 | -0.44 | -0.43 | -0.43 | -0.42 | -0.42 | -0.42 | -0.40 | -0.32 | -0.31 | -0.27 | - | 0.05⋆ |
| online-F | -0.52 | -0.49 | -0.49 | -0.48 | -0.48 | -0.47 | -0.47 | -0.47 | -0.45 | -0.37 | -0.36 | -0.32 | -0.05 | - |
| score | 0.14 | 0.11 | 0.11 | 0.11 | 0.10 | 0.09 | 0.09 | 0.09 | 0.08 | -0.01 | -0.02 | -0.06 | -0.33 | -0.38 |
| rank | 1–9 | 1–9 | 1–9 | 1–9 | 1–9 | 1–9 | 1–9 | 1–9 | 1–9 | 10–11 | 10–11 | 12 | 13 | 14 |

**Table 10:** Head to head comparison for Chinese→English systems.

|  | Tencent-ensemble | Unisound | GTCOM-Primary | Alibaba-Ens-Rerank | Alibaba-General-A | online-B | Alibaba-General-B | UMD | NICT | online-Y | online-A | uedin | online-F | online-G |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tencent-ensemble | - | 0.01 | 0.02 | 0.03 | 0.05 | 0.05⋆ | 0.05 | 0.13‡ | 0.14‡ | 0.15‡ | 0.18‡ | 0.42‡ | 0.64‡ | 0.65‡ |
| Unisound | -0.01 | - | 0.01 | 0.02 | 0.03 | 0.04 | 0.04 | 0.11‡ | 0.12‡ | 0.14‡ | 0.17‡ | 0.41‡ | 0.62‡ | 0.64‡ |
| GTCOM-Primary | -0.02 | -0.01 | - | 0.01 | 0.03 | 0.03⋆ | 0.03 | 0.11‡ | 0.12‡ | 0.13‡ | 0.16‡ | 0.40‡ | 0.62‡ | 0.63‡ |
| Alibaba-Ens-Rerank | -0.03 | -0.02 | -0.01 | - | 0.01 | 0.02 | 0.02 | 0.09‡ | 0.10‡ | 0.12‡ | 0.15‡ | 0.39‡ | 0.60‡ | 0.62‡ |
| Alibaba-General-A | -0.05 | -0.03 | -0.03 | -0.01 | - | 0.01 | 0.01 | 0.08‡ | 0.09† | 0.10‡ | 0.14‡ | 0.38‡ | 0.59‡ | 0.61‡ |
| online-B | -0.05 | -0.04 | -0.03 | -0.02 | -0.01 | - | 0.00 | 0.07† | 0.08⋆ | 0.10‡ | 0.13‡ | 0.37‡ | 0.58‡ | 0.60‡ |
| Alibaba-General-B | -0.05 | -0.04 | -0.03 | -0.02 | -0.01 | 0.00 | - | 0.07† | 0.08† | 0.10‡ | 0.13‡ | 0.37‡ | 0.58‡ | 0.60‡ |
| UMD | -0.13 | -0.11 | -0.11 | -0.09 | -0.08 | -0.07 | -0.07 | - | 0.01 | 0.03 | 0.06† | 0.30‡ | 0.51‡ | 0.53‡ |
| NICT | -0.14 | -0.12 | -0.12 | -0.10 | -0.09 | -0.08 | -0.08 | -0.01 | - | 0.01 | 0.04† | 0.28‡ | 0.50‡ | 0.52‡ |
| online-Y | -0.15 | -0.14 | -0.13 | -0.12 | -0.10 | -0.10 | -0.10 | -0.03 | -0.01 | - | 0.03 | 0.27‡ | 0.49‡ | 0.50‡ |
| online-A | -0.18 | -0.17 | -0.16 | -0.15 | -0.14 | -0.13 | -0.13 | -0.06 | -0.04 | -0.03 | - | 0.24‡ | 0.46‡ | 0.47‡ |
| uedin | -0.42 | -0.41 | -0.40 | -0.39 | -0.38 | -0.37 | -0.37 | -0.30 | -0.28 | -0.27 | -0.24 | - | 0.22‡ | 0.23‡ |
| online-F | -0.64 | -0.62 | -0.62 | -0.60 | -0.59 | -0.58 | -0.58 | -0.51 | -0.50 | -0.49 | -0.46 | -0.22 | - | 0.02 |
| online-G | -0.65 | -0.64 | -0.63 | -0.62 | -0.61 | -0.60 | -0.60 | -0.53 | -0.52 | -0.50 | -0.47 | -0.23 | -0.02 | - |
| score | 0.22 | 0.21 | 0.20 | 0.18 | 0.17 | 0.17 | 0.17 | 0.09 | 0.08 | 0.07 | 0.04 | -0.20 | -0.42 | -0.43 |
| rank | 1–7 | 1–7 | 1–7 | 1–7 | 1–7 | 1–7 | 1–7 | 8–11 | 8–11 | 8–11 | 8–11 | 12 | 13–14 | 13–14 |

**Table 11:** Head to head comparison for English→Chinese systems.

|  | CUNI-Transformer | uedin | online-B | online-A | online-G |
|---|---|---|---|---|---|
| CUNI-Transformer | - | 0.13‡ | 0.18‡ | 0.32‡ | 0.48‡ |
| uedin | -0.13 | - | 0.05⋆ | 0.19‡ | 0.35‡ |
| online-B | -0.18 | -0.05 | - | 0.14‡ | 0.30‡ |
| online-A | -0.32 | -0.19 | -0.14 | - | 0.16‡ |
| online-G | -0.48 | -0.35 | -0.30 | -0.16 | - |
| score | 0.30 | 0.17 | 0.12 | -0.02 | -0.18 |
| rank | 1 | 2 | 3 | 4 | 5 |

**Table 12:** Head to head comparison for Czech→English systems.

|  | CUNI-Transformer | uedin | online-B | online-A | online-G |
|---|---|---|---|---|---|
| CUNI-Transformer | - | 0.21‡ | 0.49‡ | 0.71‡ | 0.84‡ |
| uedin | -0.21 | - | 0.28‡ | 0.50‡ | 0.63‡ |
| online-B | -0.49 | -0.28 | - | 0.22‡ | 0.35‡ |
| online-A | -0.71 | -0.50 | -0.22 | - | 0.13‡ |
| online-G | -0.84 | -0.63 | -0.35 | -0.13 | - |
| score | 0.59 | 0.38 | 0.10 | -0.12 | -0.25 |
| rank | 1 | 2 | 3 | 4 | 5 |

**Table 13:** Head to head comparison for English→Czech systems.

| | RWTH | UCAM | NTT | ONLINE-B | MLLP-UPV | JHU | UBIQUS-NMT | ONLINE-Y | ONLINE-A | UEDIN | LMU-NMT | NJUNMT-PRIVATE | ONLINE-G | ONLINE-F | RWTH-UNSUPER | LMU-UNSUP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RWTH | - | 0.02 | 0.05⋆ | 0.07‡ | 0.09‡ | 0.10‡ | 0.10‡ | 0.10‡ | 0.15‡ | 0.15‡ | 0.25‡ | 0.26‡ | 0.49‡ | 0.71‡ | 1.17‡ | 1.25‡ |
| UCAM | -0.02 | - | 0.04 | 0.05⋆ | 0.07‡ | 0.08† | 0.08‡ | 0.08‡ | 0.13‡ | 0.13‡ | 0.23‡ | 0.25‡ | 0.47‡ | 0.69‡ | 1.15‡ | 1.23‡ |
| NTT | -0.05 | -0.04 | - | 0.01 | 0.04⋆ | 0.04⋆ | 0.04† | 0.05† | 0.09‡ | 0.10‡ | 0.20‡ | 0.21‡ | 0.43‡ | 0.66‡ | 1.11‡ | 1.19‡ |
| ONLINE-B | -0.07 | -0.05 | -0.01 | - | 0.03 | 0.03 | 0.03⋆ | 0.04 | 0.08‡ | 0.09‡ | 0.18‡ | 0.20‡ | 0.42‡ | 0.64‡ | 1.10‡ | 1.18‡ |
| MLLP-UPV | -0.09 | -0.07 | -0.04 | -0.03 | - | 0.00 | 0.01 | 0.01 | 0.05† | 0.06⋆ | 0.16‡ | 0.17‡ | 0.40‡ | 0.62‡ | 1.07‡ | 1.16‡ |
| JHU | -0.10 | -0.08 | -0.04 | -0.03 | 0.00 | - | 0.00 | 0.01 | 0.05† | 0.06† | 0.15‡ | 0.17‡ | 0.39‡ | 0.61‡ | 1.07‡ | 1.15‡ |
| UBIQUS-NMT | -0.10 | -0.08 | -0.04 | -0.03 | -0.01 | 0.00 | - | 0.01 | 0.05⋆ | 0.05 | 0.15‡ | 0.17‡ | 0.39‡ | 0.61‡ | 1.07‡ | 1.15‡ |
| ONLINE-Y | -0.10 | -0.08 | -0.05 | -0.04 | -0.01 | -0.01 | -0.01 | - | 0.04⋆ | 0.05⋆ | 0.15‡ | 0.16‡ | 0.38‡ | 0.61‡ | 1.06‡ | 1.15‡ |
| ONLINE-A | -0.15 | -0.13 | -0.09 | -0.08 | -0.05 | -0.05 | -0.05 | -0.04 | - | 0.01 | 0.11‡ | 0.12‡ | 0.34‡ | 0.56‡ | 1.02‡ | 1.10‡ |
| UEDIN | -0.15 | -0.13 | -0.10 | -0.09 | -0.06 | -0.06 | -0.05 | -0.05 | -0.01 | - | 0.10‡ | 0.11‡ | 0.34‡ | 0.56‡ | 1.01‡ | 1.10‡ |
| LMU-NMT | -0.25 | -0.23 | -0.20 | -0.18 | -0.16 | -0.15 | -0.15 | -0.15 | -0.11 | -0.10 | - | 0.01 | 0.24‡ | 0.46‡ | 0.91‡ | 1.00‡ |
| NJUNMT-PRIVATE | -0.26 | -0.25 | -0.21 | -0.20 | -0.17 | -0.17 | -0.17 | -0.16 | -0.12 | -0.11 | -0.01 | - | 0.22‡ | 0.45‡ | 0.90‡ | 0.98‡ |
| ONLINE-G | -0.49 | -0.47 | -0.43 | -0.42 | -0.40 | -0.39 | -0.39 | -0.38 | -0.34 | -0.34 | -0.24 | -0.22 | - | 0.22‡ | 0.68‡ | 0.76‡ |
| ONLINE-F | -0.71 | -0.69 | -0.66 | -0.64 | -0.62 | -0.61 | -0.61 | -0.61 | -0.56 | -0.56 | -0.46 | -0.45 | -0.22 | - | 0.46‡ | 0.54‡ |
| RWTH-UNSUPER | -1.17 | -1.15 | -1.11 | -1.10 | -1.07 | -1.07 | -1.07 | -1.06 | -1.02 | -1.01 | -0.91 | -0.90 | -0.68 | -0.46 | - | 0.08‡ |
| LMU-UNSUP | -1.25 | -1.23 | -1.19 | -1.18 | -1.16 | -1.15 | -1.15 | -1.15 | -1.10 | -1.10 | -1.00 | -0.98 | -0.76 | -0.54 | -0.08 | - |
| score | 0.41 | 0.40 | 0.36 | 0.35 | 0.32 | 0.32 | 0.32 | 0.31 | 0.27 | 0.26 | 0.16 | 0.15 | -0.07 | -0.30 | -0.75 | -0.83 |
| rank | 1–8 | 1–8 | 1–8 | 1–8 | 1–8 | 1–8 | 1–8 | 1–8 | 9–10 | 9–10 | 11–12 | 11–12 | 13 | 14 | 15 | 16 |

**Table 14:** Head to head comparison for German→English systems.

| | FACEBOOK-FAIR ⋆ | ONLINE-B | MICROSOFT-MARIAN | MMT-PRODUCTION | UCAM | NTT | KIT | ONLINE-Y | JHU | UEDIN | LMU-NMT | ONLINE-A | ONLINE-F | ONLINE-G | RWTH-UNSUPER | LMU-UNSUP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FACEBOOK-FAIR ⋆ | - | 0.09† | 0.10‡ | 0.11‡ | 0.12‡ | 0.16‡ | 0.20‡ | 0.26‡ | 0.28‡ | 0.30‡ | 0.44‡ | 0.59‡ | 1.04‡ | 1.07‡ | 1.62‡ | 1.77‡ |
| ONLINE-B | -0.09 | - | 0.01 | 0.02 | 0.02 | 0.07† | 0.11‡ | 0.16‡ | 0.18‡ | 0.21‡ | 0.35‡ | 0.50‡ | 0.95‡ | 0.98‡ | 1.53‡ | 1.68‡ |
| MICROSOFT-MARIAN | -0.10 | -0.01 | - | 0.01 | 0.01 | 0.06⋆ | 0.10† | 0.16‡ | 0.17‡ | 0.20‡ | 0.34‡ | 0.49‡ | 0.94‡ | 0.97‡ | 1.52‡ | 1.67‡ |
| MMT-PRODUCTION | -0.11 | -0.02 | -0.01 | - | 0.00 | 0.05 | 0.09⋆ | 0.14‡ | 0.16‡ | 0.19‡ | 0.33‡ | 0.48‡ | 0.92‡ | 0.95‡ | 1.51‡ | 1.66‡ |
| UCAM | -0.12 | -0.02 | -0.01 | 0.00 | - | 0.05 | 0.08⋆ | 0.14‡ | 0.16‡ | 0.19‡ | 0.32‡ | 0.48‡ | 0.92‡ | 0.95‡ | 1.50‡ | 1.66‡ |
| NTT | -0.16 | -0.07 | -0.06 | -0.05 | -0.05 | - | 0.04 | 0.10‡ | 0.11† | 0.14‡ | 0.28‡ | 0.43‡ | 0.88‡ | 0.91‡ | 1.46‡ | 1.61‡ |
| KIT | -0.20 | -0.11 | -0.10 | -0.09 | -0.08 | -0.04 | - | 0.06† | 0.08⋆ | 0.10† | 0.24‡ | 0.39‡ | 0.84‡ | 0.87‡ | 1.42‡ | 1.58‡ |
| ONLINE-Y | -0.26 | -0.16 | -0.16 | -0.14 | -0.14 | -0.10 | -0.06 | - | 0.02 | 0.04 | 0.18‡ | 0.34‡ | 0.78‡ | 0.81‡ | 1.36‡ | 1.52‡ |
| JHU | -0.28 | -0.18 | -0.17 | -0.16 | -0.16 | -0.11 | -0.08 | -0.02 | - | 0.03 | 0.16‡ | 0.32‡ | 0.76‡ | 0.79‡ | 1.34‡ | 1.50‡ |
| UEDIN | -0.30 | -0.21 | -0.20 | -0.19 | -0.19 | -0.14 | -0.10 | -0.04 | -0.03 | - | 0.14‡ | 0.29‡ | 0.74‡ | 0.77‡ | 1.32‡ | 1.47‡ |
| LMU-NMT | -0.44 | -0.35 | -0.34 | -0.33 | -0.32 | -0.28 | -0.24 | -0.18 | -0.16 | -0.14 | - | 0.15‡ | 0.60‡ | 0.63‡ | 1.18‡ | 1.33‡ |
| ONLINE-A | -0.59 | -0.50 | -0.49 | -0.48 | -0.48 | -0.43 | -0.39 | -0.34 | -0.32 | -0.29 | -0.15 | - | 0.44‡ | 0.48‡ | 1.03‡ | 1.18‡ |
| ONLINE-F | -1.04 | -0.95 | -0.94 | -0.92 | -0.92 | -0.88 | -0.84 | -0.78 | -0.76 | -0.74 | -0.60 | -0.44 | - | 0.03 | 0.58‡ | 0.74‡ |
| ONLINE-G | -1.07 | -0.98 | -0.97 | -0.95 | -0.95 | -0.91 | -0.87 | -0.81 | -0.79 | -0.77 | -0.63 | -0.48 | -0.03 | - | 0.55‡ | 0.71‡ |
| RWTH-UNSUPER | -1.62 | -1.53 | -1.52 | -1.51 | -1.50 | -1.46 | -1.42 | -1.36 | -1.34 | -1.32 | -1.18 | -1.03 | -0.58 | -0.55 | - | 0.16‡ |
| LMU-UNSUP | -1.77 | -1.68 | -1.67 | -1.66 | -1.66 | -1.61 | -1.58 | -1.52 | -1.50 | -1.47 | -1.33 | -1.18 | -0.74 | -0.71 | -0.16 | - |
| score | 0.65 | 0.56 | 0.55 | 0.54 | 0.54 | 0.49 | 0.45 | 0.40 | 0.38 | 0.35 | 0.21 | 0.06 | -0.39 | -0.42 | -0.97 | -1.12 |
| rank | 1 | 2–7 | 2–7 | 2–7 | 2–7 | 2–7 | 2–7 | 8–10 | 8–10 | 8–10 | 11 | 12 | 13–14 | 13–14 | 15 | 16 |

**Table 15:** Head to head comparison for English→German systems.

|  | TILDE-NC-NMT | NICT | TILDE-C-NMT | TILDE-C-NMT-2BT | UEDIN | TILDE-C-NMT-COMB | ONLINE-B | HY-NMT | TALP-UPC | ONLINE-A | CUNI-Kocmi | NEUROTOLGE.EE | ONLINE-G | UnsupTartu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TILDE-NC-NMT | - | 0.09‡ | 0.11‡ | 0.14‡ | 0.14‡ | 0.16‡ | 0.21‡ | 0.22‡ | 0.22‡ | 0.26‡ | 0.32‡ | 0.44‡ | 0.67‡ | 1.28‡ |
| NICT | -0.09 | - | 0.02 | 0.05⋆ | 0.05 | 0.07† | 0.12‡ | 0.13‡ | 0.13‡ | 0.18‡ | 0.23‡ | 0.36‡ | 0.58‡ | 1.19‡ |
| TILDE-C-NMT | -0.11 | -0.02 | - | 0.03 | 0.03 | 0.04 | 0.10‡ | 0.11† | 0.11‡ | 0.15‡ | 0.21‡ | 0.33‡ | 0.56‡ | 1.17‡ |
| TILDE-C-NMT-2BT | -0.14 | -0.05 | -0.03 | - | 0.00 | 0.02 | 0.07† | 0.08† | 0.08† | 0.12‡ | 0.18‡ | 0.30‡ | 0.53‡ | 1.14‡ |
| UEDIN | -0.14 | -0.05 | -0.03 | 0.00 | - | 0.02 | 0.07† | 0.08† | 0.08† | 0.12‡ | 0.18‡ | 0.30‡ | 0.53‡ | 1.14‡ |
| TILDE-C-NMT-COMB | -0.16 | -0.07 | -0.04 | -0.02 | -0.02 | - | 0.05⋆ | 0.06 | 0.06⋆ | 0.11‡ | 0.16‡ | 0.29‡ | 0.51‡ | 1.12‡ |
| ONLINE-B | -0.21 | -0.12 | -0.10 | -0.07 | -0.07 | -0.05 | - | 0.01 | 0.01 | 0.05⋆ | 0.11† | 0.23‡ | 0.46‡ | 1.07‡ |
| HY-NMT | -0.22 | -0.13 | -0.11 | -0.08 | -0.08 | -0.06 | -0.01 | - | 0.00 | 0.04⋆ | 0.10† | 0.22‡ | 0.45‡ | 1.06‡ |
| TALP-UPC | -0.22 | -0.13 | -0.11 | -0.08 | -0.08 | -0.06 | -0.01 | 0.00 | - | 0.04⋆ | 0.10† | 0.22‡ | 0.45‡ | 1.06‡ |
| ONLINE-A | -0.26 | -0.18 | -0.15 | -0.12 | -0.12 | -0.11 | -0.05 | -0.04 | -0.04 | - | 0.06 | 0.18‡ | 0.40‡ | 1.01‡ |
| CUNI-Kocmi | -0.32 | -0.23 | -0.21 | -0.18 | -0.18 | -0.16 | -0.11 | -0.10 | -0.10 | -0.06 | - | 0.12‡ | 0.35‡ | 0.96‡ |
| NEUROTOLGE.EE | -0.44 | -0.36 | -0.33 | -0.30 | -0.30 | -0.29 | -0.23 | -0.22 | -0.22 | -0.18 | -0.12 | - | 0.22‡ | 0.83‡ |
| ONLINE-G | -0.67 | -0.58 | -0.56 | -0.53 | -0.53 | -0.51 | -0.46 | -0.45 | -0.45 | -0.40 | -0.35 | -0.22 | - | 0.61‡ |
| UnsupTartu | -1.28 | -1.19 | -1.17 | -1.14 | -1.14 | -1.12 | -1.07 | -1.06 | -1.06 | -1.01 | -0.96 | -0.83 | -0.61 | - |
| score | 0.33 | 0.24 | 0.21 | 0.19 | 0.19 | 0.17 | 0.12 | 0.11 | 0.11 | 0.06 | 0.01 | -0.12 | -0.34 | -0.95 |
| rank | 1 | 2–9 | 2–9 | 2–9 | 2–9 | 2–9 | 2–9 | 2–9 | 2–9 | 10–11 | 10–11 | 12 | 13 | 14 |

**Table 16:** Head to head comparison for Estonian→English systems.

|  | TILDE-NC-NMT | NICT | TILDE-C-NMT | TILDE-C-NMT-2BT | Aalto | HY-NMT | UEDIN | CUNI-Kocmi | TALP-UPC | ONLINE-B | NEUROTOLGE.EE | ONLINE-A | ONLINE-G | PARFDA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TILDE-NC-NMT | - | 0.10⋆ | 0.12† | 0.13‡ | 0.21‡ | 0.22‡ | 0.25‡ | 0.33‡ | 0.37‡ | 0.45‡ | 0.68‡ | 0.74‡ | 0.95‡ | 1.07‡ |
| NICT | -0.10 | - | 0.03 | 0.03 | 0.11† | 0.12† | 0.16‡ | 0.24‡ | 0.27‡ | 0.36‡ | 0.58‡ | 0.65‡ | 0.86‡ | 0.97‡ |
| TILDE-C-NMT | -0.12 | -0.03 | - | 0.01 | 0.09⋆ | 0.10⋆ | 0.13† | 0.21‡ | 0.25‡ | 0.33‡ | 0.56‡ | 0.62‡ | 0.83‡ | 0.95‡ |
| TILDE-C-NMT-2BT | -0.13 | -0.03 | -0.01 | - | 0.08⋆ | 0.09⋆ | 0.12† | 0.20‡ | 0.24‡ | 0.32‡ | 0.55‡ | 0.61‡ | 0.82‡ | 0.94‡ |
| Aalto | -0.21 | -0.11 | -0.09 | -0.08 | - | 0.01 | 0.05 | 0.12† | 0.16‡ | 0.24‡ | 0.47‡ | 0.54‡ | 0.75‡ | 0.86‡ |
| HY-NMT | -0.22 | -0.12 | -0.10 | -0.09 | -0.01 | - | 0.03 | 0.11† | 0.15‡ | 0.23‡ | 0.46‡ | 0.52‡ | 0.73‡ | 0.85‡ |
| UEDIN | -0.25 | -0.16 | -0.13 | -0.12 | -0.05 | -0.03 | - | 0.08⋆ | 0.11† | 0.20‡ | 0.43‡ | 0.49‡ | 0.70‡ | 0.81‡ |
| CUNI-Kocmi | -0.33 | -0.24 | -0.21 | -0.20 | -0.12 | -0.11 | -0.08 | - | 0.04 | 0.12† | 0.35‡ | 0.41‡ | 0.62‡ | 0.74‡ |
| TALP-UPC | -0.37 | -0.27 | -0.25 | -0.24 | -0.16 | -0.15 | -0.11 | -0.04 | - | 0.08⋆ | 0.31‡ | 0.38‡ | 0.59‡ | 0.70‡ |
| ONLINE-B | -0.45 | -0.36 | -0.33 | -0.32 | -0.24 | -0.23 | -0.20 | -0.12 | -0.08 | - | 0.23‡ | 0.29‡ | 0.50‡ | 0.62‡ |
| NEUROTOLGE.EE | -0.68 | -0.58 | -0.56 | -0.55 | -0.47 | -0.46 | -0.43 | -0.35 | -0.31 | -0.23 | - | 0.06⋆ | 0.27‡ | 0.39‡ |
| ONLINE-A | -0.74 | -0.65 | -0.62 | -0.61 | -0.54 | -0.52 | -0.49 | -0.41 | -0.38 | -0.29 | -0.06 | - | 0.21‡ | 0.32‡ |
| ONLINE-G | -0.95 | -0.86 | -0.83 | -0.82 | -0.75 | -0.73 | -0.70 | -0.62 | -0.59 | -0.50 | -0.27 | -0.21 | - | 0.11‡ |
| PARFDA | -1.07 | -0.97 | -0.95 | -0.94 | -0.86 | -0.85 | -0.81 | -0.74 | -0.70 | -0.62 | -0.39 | -0.32 | -0.11 | - |
| score | 0.55 | 0.45 | 0.43 | 0.42 | 0.34 | 0.33 | 0.29 | 0.22 | 0.18 | 0.10 | -0.13 | -0.20 | -0.41 | -0.52 |
| rank | 1 | 2–4 | 2–4 | 2–4 | 5–7 | 5–7 | 5–7 | 8–9 | 8–9 | 10 | 11 | 12 | 13 | 14 |

**Table 17:** Head to head comparison for English→Estonian systems.

|  | NICT | HY-NMT | UEDIN | CUNI-Kocmi | ONLINE-B | TALP-UPC | ONLINE-A | ONLINE-G | JUCBNMT |
|---|---|---|---|---|---|---|---|---|---|
| NICT | - | 0.02 | 0.05 | 0.07† | 0.07† | 0.11‡ | 0.11‡ | 0.29‡ | 0.56‡ |
| HY-NMT | -0.02 | - | 0.03 | 0.05⋆ | 0.05⋆ | 0.08† | 0.08† | 0.26‡ | 0.53‡ |
| UEDIN | -0.05 | -0.03 | - | 0.02 | 0.02 | 0.06† | 0.06† | 0.24‡ | 0.51‡ |
| CUNI-Kocmi | -0.07 | -0.05 | -0.02 | - | 0.00 | 0.04 | 0.04 | 0.22‡ | 0.49‡ |
| ONLINE-B | -0.07 | -0.05 | -0.02 | 0.00 | - | 0.03 | 0.03 | 0.21‡ | 0.48‡ |
| TALP-UPC | -0.11 | -0.08 | -0.06 | -0.04 | -0.03 | - | 0.00 | 0.18‡ | 0.45‡ |
| ONLINE-A | -0.11 | -0.08 | -0.06 | -0.04 | -0.03 | 0.00 | - | 0.18‡ | 0.45‡ |
| ONLINE-G | -0.29 | -0.26 | -0.24 | -0.22 | -0.21 | -0.18 | -0.18 | - | 0.27‡ |
| JUCBNMT | -0.56 | -0.53 | -0.51 | -0.49 | -0.48 | -0.45 | -0.45 | -0.27 | - |
| score | 0.15 | 0.13 | 0.10 | 0.08 | 0.08 | 0.05 | 0.04 | -0.13 | -0.40 |
| rank | 1–7 | 1–7 | 1–7 | 1–7 | 1–7 | 1–7 | 1–7 | 8 | 9 |

**Table 18:** Head to head comparison for Finnish→English systems.

|  | NICT | HY-NMT | uedin | Aalto | HY-NMT-2step | talp-upc | CUNI-Kocmi | online-B | online-A | online-G | HY-SMT | HY-AH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NICT | - | 0.05 | 0.20‡ | 0.25‡ | 0.26‡ | 0.28‡ | 0.34‡ | 0.34‡ | 0.73‡ | 0.75‡ | 0.86‡ | 0.89‡ |
| HY-NMT | -0.05 | - | 0.14† | 0.19‡ | 0.21‡ | 0.23‡ | 0.28‡ | 0.28‡ | 0.68‡ | 0.70‡ | 0.80‡ | 0.83‡ |
| uedin | -0.20 | -0.14 | - | 0.05 | 0.07 | 0.09⋆ | 0.14† | 0.14‡ | 0.54‡ | 0.56‡ | 0.66‡ | 0.69‡ |
| Aalto | -0.25 | -0.19 | -0.05 | - | 0.01 | 0.03 | 0.09⋆ | 0.09⋆ | 0.48‡ | 0.50‡ | 0.61‡ | 0.64‡ |
| HY-NMT-2step | -0.26 | -0.21 | -0.07 | -0.01 | - | 0.02 | 0.07 | 0.07⋆ | 0.47‡ | 0.49‡ | 0.59‡ | 0.63‡ |
| talp-upc | -0.28 | -0.23 | -0.09 | -0.03 | -0.02 | - | 0.05 | 0.05 | 0.45‡ | 0.47‡ | 0.57‡ | 0.61‡ |
| CUNI-Kocmi | -0.34 | -0.28 | -0.14 | -0.09 | -0.07 | -0.05 | - | 0.00 | 0.40‡ | 0.42‡ | 0.52‡ | 0.55‡ |
| online-B | -0.34 | -0.28 | -0.14 | -0.09 | -0.07 | -0.05 | 0.00 | - | 0.39‡ | 0.42‡ | 0.52‡ | 0.55‡ |
| online-A | -0.73 | -0.68 | -0.54 | -0.48 | -0.47 | -0.45 | -0.40 | -0.39 | - | 0.02 | 0.12† | 0.16‡ |
| online-G | -0.75 | -0.70 | -0.56 | -0.50 | -0.49 | -0.47 | -0.42 | -0.42 | -0.02 | - | 0.10† | 0.14‡ |
| HY-SMT | -0.86 | -0.80 | -0.66 | -0.61 | -0.59 | -0.57 | -0.52 | -0.52 | -0.12 | -0.10 | - | 0.03 |
| HY-AH | -0.89 | -0.83 | -0.69 | -0.64 | -0.63 | -0.61 | -0.55 | -0.55 | -0.16 | -0.14 | -0.03 | - |
| score | 0.52 | 0.47 | 0.32 | 0.27 | 0.26 | 0.24 | 0.18 | 0.18 | -0.21 | -0.23 | -0.33 | -0.37 |
| rank | 1–2 | 1–2 | 3–8 | 3–8 | 3–8 | 3–8 | 3–8 | 3–8 | 9–10 | 9–10 | 11–12 | 11–12 |

**Table 19:** Head to head comparison for English→Finnish systems.

|  | Alibaba | online-B | online-G | uedin | online-A | afrl-syscomb | JHU | online-F |
|---|---|---|---|---|---|---|---|---|
| Alibaba | - | 0.02 | 0.04 | 0.10‡ | 0.18‡ | 0.23‡ | 0.24‡ | 0.61‡ |
| online-B | -0.02 | - | 0.02 | 0.08⋆ | 0.16‡ | 0.21‡ | 0.22‡ | 0.59‡ |
| online-G | -0.04 | -0.02 | - | 0.06⋆ | 0.14‡ | 0.18‡ | 0.20‡ | 0.57‡ |
| uedin | -0.10 | -0.08 | -0.06 | - | 0.08† | 0.12‡ | 0.14‡ | 0.51‡ |
| online-A | -0.18 | -0.16 | -0.14 | -0.08 | - | 0.05⋆ | 0.06⋆ | 0.43‡ |
| afrl-syscomb | -0.23 | -0.21 | -0.18 | -0.12 | -0.05 | - | 0.01 | 0.38‡ |
| JHU | -0.24 | -0.22 | -0.20 | -0.14 | -0.06 | -0.01 | - | 0.37‡ |
| online-F | -0.61 | -0.59 | -0.57 | -0.51 | -0.43 | -0.38 | -0.37 | - |
| score | 0.21 | 0.19 | 0.17 | 0.11 | 0.03 | -0.01 | -0.03 | -0.40 |
| rank | 1–3 | 1–3 | 1–3 | 4 | 5 | 6–7 | 6–7 | 8 |

**Table 20:** Head to head comparison for Russian→English systems.

|  | Alibaba-Ens | online-G | online-B | uedin | PROMT-Hyb-Marian | PROMT-Hyb-OpenNMT | online-A | PROMT-Rule-based | online-F |
|---|---|---|---|---|---|---|---|---|---|
| Alibaba-Ens | - | 0.03 | 0.19‡ | 0.21‡ | 0.24‡ | 0.29‡ | 0.36‡ | 0.43‡ | 0.93‡ |
| online-G | -0.03 | - | 0.16‡ | 0.18‡ | 0.21‡ | 0.26‡ | 0.33‡ | 0.40‡ | 0.90‡ |
| online-B | -0.19 | -0.16 | - | 0.01 | 0.04⋆ | 0.09‡ | 0.16‡ | 0.23‡ | 0.74‡ |
| uedin | -0.21 | -0.18 | -0.01 | - | 0.03 | 0.08† | 0.15‡ | 0.22‡ | 0.72‡ |
| PROMT-Hyb-Marian | -0.24 | -0.21 | -0.04 | -0.03 | - | 0.05⋆ | 0.12‡ | 0.19‡ | 0.69‡ |
| PROMT-Hyb-OpenNMT | -0.29 | -0.26 | -0.09 | -0.08 | -0.05 | - | 0.07† | 0.14‡ | 0.65‡ |
| online-A | -0.36 | -0.33 | -0.16 | -0.15 | -0.12 | -0.07 | - | 0.07† | 0.58‡ |
| PROMT-Rule-based | -0.43 | -0.40 | -0.23 | -0.22 | -0.19 | -0.14 | -0.07 | - | 0.50‡ |
| online-F | -0.93 | -0.90 | -0.74 | -0.72 | -0.69 | -0.65 | -0.58 | -0.50 | - |
| score | 0.35 | 0.32 | 0.16 | 0.14 | 0.12 | 0.07 | -0.00 | -0.07 | -0.58 |
| rank | 1–2 | 1–2 | 3–5 | 3–5 | 3–5 | 6 | 7 | 8 | 9 |

**Table 21:** Head to head comparison for English→Russian systems.

|  | online-G | online-A | Alibaba-Ens | online-B | uedin | NICT |
|---|---|---|---|---|---|---|
| online-G | - | 0.02 | 0.06⋆ | 0.06† | 0.11‡ | 0.13‡ |
| online-A | -0.02 | - | 0.04 | 0.04 | 0.08⋆ | 0.10‡ |
| Alibaba-Ens | -0.06 | -0.04 | - | 0.01 | 0.05 | 0.07† |
| online-B | -0.06 | -0.04 | -0.01 | - | 0.04 | 0.06⋆ |
| uedin | -0.11 | -0.08 | -0.05 | -0.04 | - | 0.02 |
| NICT | -0.13 | -0.10 | -0.07 | -0.06 | -0.02 | - |
| score | 0.09 | 0.07 | 0.03 | 0.02 | -0.02 | -0.04 |
| rank | 1–6 | 1–6 | 1–6 | 1–6 | 1–6 | 1–6 |

**Table 22:** Head to head comparison for Turkish→English systems.

|  | online-B | uedin | Alibaba-Ens-A | NICT | Alibaba-Ens-B | online-G | RWTH | online-A |
|---|---|---|---|---|---|---|---|---|
| online-B | - | 0.05 | 0.06 | 0.15⋆ | 0.17† | 0.22‡ | 0.34‡ | 0.53‡ |
| uedin | -0.05 | - | 0.01 | 0.09⋆ | 0.11⋆ | 0.16† | 0.28‡ | 0.48‡ |
| Alibaba-Ens-A | -0.06 | -0.01 | - | 0.09 | 0.10 | 0.16† | 0.28‡ | 0.47‡ |
| NICT | -0.15 | -0.09 | -0.09 | - | 0.02 | 0.07 | 0.19‡ | 0.38‡ |
| Alibaba-Ens-B | -0.17 | -0.11 | -0.10 | -0.02 | - | 0.05 | 0.17† | 0.36‡ |
| online-G | -0.22 | -0.16 | -0.16 | -0.07 | -0.05 | - | 0.12⋆ | 0.31‡ |
| RWTH | -0.34 | -0.28 | -0.28 | -0.19 | -0.17 | -0.12 | - | 0.19‡ |
| online-A | -0.53 | -0.48 | -0.47 | -0.38 | -0.36 | -0.31 | -0.19 | - |
| score | 0.28 | 0.22 | 0.22 | 0.13 | 0.11 | 0.06 | -0.06 | -0.25 |
| rank | 1–6 | 1–6 | 1–6 | 1–6 | 1–6 | 1–6 | 7 | 8 |

**Table 23:** Head to head comparison for English→Turkish systems.

|  | CUNI-Transformer | uedin | newstest2018-ref | online-B | online-A | online-G |
|---|---|---|---|---|---|---|
| CUNI-Transformer | - | 0.15‡ | 0.18‡ | 0.54‡ | 0.85‡ | 1.02‡ |
| uedin | -0.15 | - | 0.04 | 0.39‡ | 0.70‡ | 0.88‡ |
| newstest2018-ref | -0.18 | -0.04 | - | 0.36‡ | 0.66‡ | 0.84‡ |
| online-B | -0.54 | -0.39 | -0.36 | - | 0.31‡ | 0.48‡ |
| online-A | -0.85 | -0.70 | -0.66 | -0.31 | - | 0.18‡ |
| online-G | -1.02 | -0.88 | -0.84 | -0.48 | -0.18 | - |
| score | 0.67 | 0.52 | 0.48 | 0.13 | -0.18 | -0.35 |
| rank | 1 | 2–3 | 2–3 | 4 | 5 | 6 |

**Table 24:** Head to head comparison for Czech→English systems, based on source-based DA.

# Findings of the Third Shared Task on Multimodal Machine Translation

**Loïc Barrault**[1], **Fethi Bougares**[1], **Lucia Specia**[2],
**Chiraag Lala**[2], **Desmond Elliott**[3] and **Stella Frank**[4]
[1]LIUM, University of Le Mans
[2]Department of Computer Science, University of Sheffield
[3]Department of Computer Science, University of Copenhagen
[4]Centre for Language Evolution, University of Edinburgh
loic.barrault@univ-lemans.fr

## Abstract

We present the results from the third shared task on multimodal machine translation. In this task a source sentence in English is supplemented by an image and participating systems are required to generate a translation for such a sentence into German, French or Czech. The image can be used in addition to (or instead of) the source sentence. This year the task was extended with a third target language (Czech) and a new test set. In addition, a variant of this task was introduced with its own test set where the source sentence is given in multiple languages: English, French and German, and participating systems are required to generate a translation in Czech. Seven teams submitted 45 different systems to the two variants of the task. Compared to last year, the performance of the multimodal submissions improved, but text-only systems remain competitive.

## 1 Introduction

The Shared Task on Multimodal Machine Translation tackles the problem of generating a description of an image in a target language using the image itself and its English description. This task can be addressed as either a pure translation task from the source English descriptions (ignoring the corresponding image), or as a multimodal translation task where the translation process is guided by the image in addition to the source description.

Initial results in this area showed the potential for visual context to improve translation quality (Elliott et al., 2015; Hitschler et al., 2016). This was followed by a wide range of work in the first two editions of this shared task at the WMT in 2016 and 2017 (Specia et al., 2016; Elliott et al., 2017).

This year we challenged participants to target the task of multimodal translation, with two variants:

- Task 1: **Multimodal translation** takes an image with a source language description that is then translated into a target language. The training data consists of source-target parallel sentences and their corresponding images.

- Task 1b: **Multisource multimodal translation** takes an image with a description in three source languages that is then translated into a target language. The training data consists of source-target parallel data and their corresponding images, but where the source sentences are presented in three different languages, all parallel.

Task 1 is identical to previous editions of the shared task, however, it now includes an additional Czech target language. Therefore, participants can submit translations to any of the following languages: German, French and Czech. This extension means the Multi30K dataset (Elliott et al., 2016) is now 5-way aligned, with images described in English, which are translated into German, French and Czech.[1] Task 1b is similar to Task 1; the main difference is that multiple source languages can be used (simultaneously) and Czech is the only target language.

We introduce two new evaluation sets that extend the existing Multi30K dataset: a set of 1071 English sentences and their corresponding images and translations for Task 1, and 1,000 translations for the 2017 test set into Czech for Task 1b.

Another new feature of this year's shared task is the introduction of a new evaluation metric: Lexical Translation Accuracy (LTA), which measures

[1] The current version of the dataset can be found here: https://github.com/multi30k/dataset

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 304–323
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64029

the accuracy of a system at translating correctly a subset of ambiguous source language words.

Participants could submit both constrained (shared task data only) and unconstrained (any data) systems for both tasks, with a limit of two systems per task variant and language pair per team.

## 2 Datasets

The Multi30K dataset (Elliott et al., 2016) is the primary resource for the shared task. It contains 31K images originally described in English (Young et al., 2014) with two types of multilingual data: a collection of professionally translated German sentences, and a collection of independently crowd-sourced German descriptions.

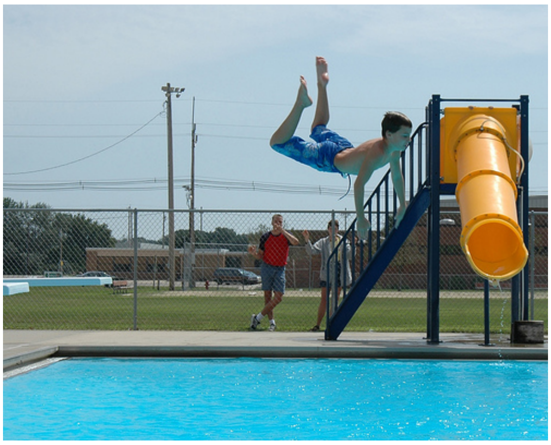
Over the two last years, we have extended the Multi30K dataset with 2,071 new images and two additional languages for the translation task: French and Czech. Table 1 presents an overview of the new evaluation datasets. Figure 1 shows an example of an image with an aligned English-German-French-Czech description.

This year we also released a new version of the evaluation datasets featuring a subset of sentences that contain ambiguous source language words, which may have different senses in the target language. We expect that these ambiguous words could benefit from additional visual context.

In addition to releasing the parallel text, we also distributed two types of visual features extracted from a pre-trained ResNet-50 object recognition model (He et al., 2016) for all of the images, namely the 'res4_relu' convolutional features (which preserve the spatial location of a feature in the original image) and averaged pooled features.

### Multi30K Czech Translations

This year the Multi30K dataset was extended with translations of the image descriptions into Czech. The translations were produced by 15 workers (university and high school students and teachers, all with a good command of English) at the cost of EUR 3,500. The translators used the same platform that was used to collect the French translations for the Multi30K dataset. The Czech translators had access to the source segment in English and the image only (no automatic translation into Czech was presented). The translated segments were automatically checked for mismatching punctuation, spelling errors (using `aspell`), inadequately short and long sentences, and non-standard characters. The segments containing errors were manually checked and fixed if needed. In total, 5,255 translated segments (16%) were corrected. After the manual correction, 1% of the segments were sampled and manually annotated for translation quality. This annotation task was performed by three annotators (and every segment was annotated by two different people to measure annotation agreement). We found that 94% of the segments did not contain any spelling errors, 96% of the segments fully preserved the meaning, and 75% of translations were annotated as fluent Czech. The remaining 25% contained some stylistic problems (usually inappropriate lexical choice and/or word order adopted from the English source segment). However, the annotation agreement for stylistic problems was substantially lower compared to other categories due to the subjectivity of deciding on the best style for a translation.



En: A boy dives into a pool near a water slide.
De: Ein Junge taucht in der Nähe einer Wasserrutsche in ein Schwimmbecken.
Fr: Un garçon plonge dans une piscine près d'un toboggan.
Cs: Chlapec skáče do bazénu poblíž skluzavky.

Figure 1: Example of an image with a source description in English, together with its German, French and Czech translations.

### Test 2018 dataset

As our new evaluation data for Task 1, we collected German, French and Czech translations for the test set used in the 2017 edition of the Multilingual Image Description Generation task, which only contained English descriptions. This test set contains images from five of the six Flickr groups used to create the original Flickr30K dataset[2]. We

[2] Strangers!, Wild Child, Dogs in Action, Action Photography, and Outdoor Activities.

| | Training set | Development set | Test set 2018 - Task 1 | Test set 2018 - Task 1b |
|---|---|---|---|---|
| Instances | 29,000 | 1,014 | 1071 | 1,000 |

Table 1: Overview of the Multi30K training, development and 2018 test datasets. The figures correspond to tuples with an image and parallel sentences in four languages: English, German, French and Czech.

| Group | Task 1 | Task 1b |
|---|---|---|
| Strangers! | 154 | 150 |
| Wild Child | 83 | 83 |
| Dogs in Action | 92 | 78 |
| Action Photography | 259 | 238 |
| Flickr Social Club | 263 | 241 |
| Everything Outdoor | 214 | 206 |
| Outdoor Activities | 6 | 4 |

Table 2: Distribution of images in the Test 2018 dataset by Flickr group.

sampled additional images from two thematically related groups (Everything Outdoor and Flickr Social Club) because Outdoor Activities only returned 10 new CC-licensed images and Flickr-Social no longer exists. The translations were collected using the same procedure as before for each of the languages: professional translations for German and internally crowdsourced translations for French and Czech (see (Elliott et al., 2017)), as described above. The new evaluation data for Task 1b consists of Czech translations, which we collected following the procedure described above. Table 2 shows the distribution of images across the groups and tasks. We initially downloaded 2,000 images per Flickr group, which were then manually filtered by three of the authors. The filtering was done to remove (near) duplicate images, clearly watermarked images, and images with dubious content. This process resulted in a total of 2,071 images, 1,000 were used for Task 1 and 1,071 for Task 1b.

### Dataset for LTA

In this year's task we also evaluate systems using Lexical Translation Accuracy (LTA) (Lala and Specia, 2018). LTA measures how accurately a system translates a subset of ambiguous words found in the Multi30K corpus. To measure this accuracy, we extract a subset of triplets form the Multi30K dataset in the form $(i, aw, clt)$ where $i$ is the index representing an instance in the test set, $aw$ is an ambiguous word in English found in that instance $i$, and $clt$ is the set of correct lexical translations of $aw$ in the target language that conform to the context $i$. A word is said to be ambiguous in the source language if it has multiple translations (as given in the Multi30K corpus) with different meanings.

We prepared the evaluation dataset following the procedure described in Lala and Specia (2018), with some additional steps. First, the parallel text in the Multi30K training and the validation sets are decompounded with SECOS (Riedl and Biemann, 2016) (for German only) and lemmatised[3]. Second, we perform automatic word alignment using fast_align (Dyer et al., 2013) to identify the English words that are aligned to two or more different words in the target language. This step results in a dictionary of $\{key : val\}$ pairs, where $key$ is a potentially ambiguous English word, and $val$ is the set of words in the target language that align to $key$. This dictionary is then filtered by humans, students of translation studies who are fluent in both the source and target languages, to remove incorrect/noisy alignments and unambiguous instances, resulting in a cleaned dictionary containing $\{aw : lt\}$ pairs, where $aw$ is an ambiguous English word, and $lt$ is the set of lexical translations of $aw$ in the corpus. For English-Czech, we were unable to perform this 'human filtering' step, and so we use the unfiltered, noisy dictionary. Table 3 shows summary statistics about number of ambiguous words and the total number of their instances in the training and validation sets.

Given a dictionary, we identify instances $i$ in the test sets[4] which contain an ambiguous word $aw$ from the dictionary, resulting in triplets of the form $(i, aw, lt)$. At this stage we again involve human

[3] For English, German and French, we use the tool from `http://staffwww.dcs.shef.ac.uk/people/A.Aker/activityNLPProjects.html`. For Czech, we pre-processed the data using MorphoDiTa (Straková et al., 2014) from `http://ufal.mff.cuni.cz/morphodita`

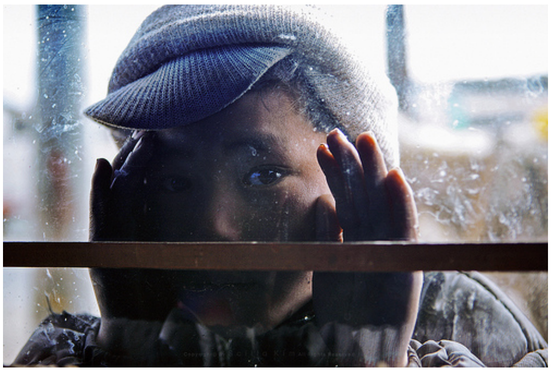
[4] The test data and the submissions undergo the same pre-processing steps as the training and the validation sets.

| Language Pair | Ambiguous Words | Instances |
|---|---|---|
| EN-DE | 745 | 53,868 |
| EN-FR | 661 | 44,779 |
| EN-CS | 3217 | 187,495 |

Table 3: Statistics of the ambiguous words extracted from the training and validation sets after human filtering (dictionary filtering). For EN-CS, the numbers are larger because we could not perform the dictionary filtering step.

annotators (students of translation studies) to select, from the set of lexical translations $lt$, only those translations, denoted as $clt$, which conform to the source context $i$ - both image and its English description. For example, in the test instance shown in Figure 2, *hat* is an ambiguous word $aw$ and {*kappe, mütze, hüten, kopf, kopfbedeckung, kopfbedeckungen, hut, helm, hüte, helmen, mützen*} is the set of its lexical translations $lt$. The human annotator looked at both the image and its description and then selected the following subset {*kappe, mütze, mützen*} as the correct lexical translations $clt$ that conform to the context of the test instance in Figure 2. We also asked annotators to expand the *clt* set with other synonyms outside the *lt* set that satisfy the context if they can. The number of ambiguous words and instances for each language pair in the resulting dataset for the test instances is given in Table 4. For English-Czech, while the first human filtering step (dictionary filtering) was not performed, the second human filtering step (test set filtering) was done. We note that this cleaning done by the Czech-English annotators was very selective, most likely due to the noisier nature of the initial annotations from the unfiltered dictionary.

Given a human filtered dictionary, the LTA evaluation is straight forward: for each MT system submission, we check if any word in *clt* is found in the translation of the submission's $i^{th}$ instance.The preprocessing steps may result in mismatches due to sub-optimal handling of morphological variants, but we do not expect this to be a rare event because the dictionaries, gold standard text, and system submissions are pre-processed using the same tools.



*En*: a cute boy with his **hat** looking out of a window.
*De*: ein süß jung mit mütze blicken aus einem fenster.
*aw*: **hat**
*lt*: {kappe, mütze, hüten, kopf, kopfbedeckung, kopfbedeckungen, hut, helm, hüte, helmen, mützen}
*clt*: {kappe, mütze, mützen}

Figure 2: A test instance with ambiguous word *aw* and lexical translation options *lt*. Human annotator corrects/selects those options *clt* which conform to the source sentence *En* and corresponding image.

| Language Pair | Ambiguous Words | Test instances |
|---|---|---|
| EN-DE | 38 | 358 |
| EN-FR | 70 | 438 |
| EN-CS | 29 | 140 |
| EN-CS(1B) | 28 | 52 |

Table 4: Statistics of dataset used for the LTA evaluation after human filtering.

## 3 Participants

This year we attracted submissions from seven groups. Table 5 presents an overview of the groups and their submission identifiers.

**AFRL-OHIO-STATE** (Task 1)

The AFRL-OHIO-STATE team builds on their previous year Visual Machine Translation (VMT) submission by combining it with text-only translation models. Two types of models were submitted: AFRL-OHIO-STATE_1_2IMPROVE_U is a system combination of the VMT system and an instantiation of a Marian NMT model (Junczys-Dowmunt et al., 2018), and AFRL-OHIO-STATE_1_4COMBO_U is a systems combination of the VMT system along with instantiations of Marian, OpenNMT, and Moses (Koehn et al., 2007).

**CUNI** (Task 1)

The CUNI submissions use two architectures based on the self-attentive Transformer model (Vaswani et al., 2017). For German and Czech, a language model is used to extract pseudo-in-

| ID | Participating team |
|---:|---|
| AFRL-OHIOSTATE | Air Force Research Laboratory & Ohio State University (Gwinnup et al., 2018) |
| CUNI | Univerzita Karlova v Praze (Helcl et al., 2018) |
| LIUMCVC | Laboratoire d'Informatique de l'Université du Maine & Universitat Autonoma de Barcelona Computer Vision Center (Caglayan et al., 2018) |
| MeMAD | Aalto University, Helsinki University & EURECOM (Grönroos et al., 2018) |
| OSU-BAIDU | Oregon State University & Baidu Research (Zheng et al., 2018) |
| SHEF | University of Sheffield (Lala et al., 2018) |
| UMONS | Université de Mons (Delbrouck and Dupont, 2018) |

Table 5: Participants in the WMT18 multimodal machine translation shared task.

domain data from all available parallel corpora and mix it with the original Multi30k data and the EU Bookshop corpus. At inference time, both submitted models use only the text input. The first model was trained using the parallel data only. The second model is a reimplementation of the Imagination model (Elliott and Kádár, 2017) adapted to the Transformer architecture. During training, the model uses the encoder states to predict the image representation. This allows using additional English-only captions from the MSCOCO dataset (Lin et al., 2014).

**LIUMCVC** (Task 1)
LIUMCVC proposes a refined version of their multimodal attention model (Caglayan et al., 2016), where source-side information from the textual encoder (i.e. last hidden state of the bidirectional gated recurrent units (GRU)) is now used to filter the convolutional feature maps before the actual decoder-side multimodal attention is computed. The authors also experiment with the impact of $L_2$ normalisation and input image size for convolutional feature extraction process and found that multimodal attention without $L_2$ normalisation performs significantly worse than baseline NMT.

**MeMAD** (Task 1)
The MeMAD team adapts the Transformer neural machine translation architecture to a multimodal setting. They use global image features extracted from Detectron (Girshick et al., 2018), a pre-trained object detection and localisation neural network, and two additional training corpora: MS-COCO (Lin et al., 2014) (an English multimodal dataset, which they extend with synthetic multilingual data) and OpenSubtitles (Lison and Tiedemann, 2016) (a multilingual, text-only dataset). Their experiments show that the effect of the visual features in the system is small; the largest differences in quality amongst the systems tested is attributed to the quality of the underlying text-only neural MT system.

**OSU-BAIDU** (Tasks 1 and 1b)
For Task 1, the OREGONSTATE system ensembles models including some neural machine translation models which only consider text information and multimodal machine translation models which also consider image information. Both types of models use global attention mechanism to align source to target words. For the multimodal model, 1024 dimensional vectors are extracted as image information from a ResNet-101 convolutional neural network and these are used to initialize the decoder. The models are trained using scheduled sampling (Bengio et al., 2015) and reinforcement learning (Rennie et al., 2017) to further improve performance.

For Task 1b, for each language in the multisource inputs, single-source models are trained using the same architecture as in Task 1. The resulting models are ensembled with different combinations. The final submissions only ensemble models trained from English-to-Czech pair, which outperforms other combinations on the development set.

**SHEF** (Tasks 1 and 1b)
For Task 1, SHEF adopts a two-step pipeline approach. In the first (base) step – submitted as a baseline system – they use an ensemble of standard attentive text-only neural machine translation models built using the NMTPY toolkit (Caglayan et al., 2017) to produce 10-best high quality trans-

lation candidates. In the second (re-ranking) step, the 10-best candidates are re-ranked using word sense disambiguation (WSD) approaches: (i) most frequency sense (MFS), (ii) lexical translation (LT) and, (iii) multimodal lexical translation (MLT). Models (i) and (ii) are baselines, whilst MLT is a novel multimodal cross-lingual WSD model. The main idea is to have the cross-lingual WSD model select the translation candidate which correctly disambiguates ambiguous words in the source sentence and the intuition is that the image could help in the disambiguation process. The re-ranking cross-lingual WSD models are based on neural sequence learning models for WSD (Raganato et al., 2017; Yuan et al., 2016) trained on the Multimodal Lexical Translation Dataset (Lala and Specia, 2018). More specifically, they train LSTMs as taggers to disambiguate/translate every word in the source sentence.

For Task 1b, the SHEF team explores three approaches. The first approach takes the concatenation of the 10-best translation candidates of German-Czech, French-Czech and English-Czech neural MT systems and then re-ranks them using the same multimodal cross-lingual WSD model as in Task 1. The second approach explores consensus between the different 10-best lists. The best hypothesis is selected according to the number of times it appears in the different n-bests. The highest ranked hypothesis with the majority votes is selected. The third approach uses data augmentation: extra source (Czech) data is generated by building systems that translate from German into English and French into English. An English-Czech neural machine translation system is then built and the 10-best list is generated. For re-ranking, classifiers are trained to predict binary scores derived from Meteor for each hypothesis in the 10-best list using word embeddings and image features.

**UMONS** (Task 1)

The UMONS submission uses as baseline a conditional GRU decoder. The architecture is enhanced with another GRU that receives as input the global visual features provided by the task (i.e. 2048-dimensional ResNet pool5 features) as well as the hidden state of the second GRU. Each GRU disposes of 256 computational units. All non-linear transformations in the decoder (apart from the textual attention module) use gated hyperbolic tangent activations. Both visual and textual representation are separately projected onto a vocabulary-sized space. At every timestep, the decoder ends up with two modality-dependent probability distributions over the target tokens, eventually merged with an element-wise addition.

**Baseline** (Tasks 1 and 1b) The baseline system for both tasks is a text-only neural machine translation system built with the NMTPY (Caglayan et al., 2017) following a standard attentive approach (Bahdanau et al., 2015) with a conditional GRU decoder. The baseline was trained using the Adam optimizer, with a learning rate of 5e$^{-5}$ and a batch size of 64. The input embedding dimensionality was set to 128 and the remainder of the hyperparameters were kept as default. Bite-pair encoding with 10,000 merge operations was used for all language pairs. For Task 1b, only the English-Czech portion of the training corpus is used.

## 4 Automatic Metric Results

The submissions were evaluated against either professional or crowd-sourced references. All submissions and references were pre-processed to lowercase, normalise punctuation, and tokenise the sentences using the Moses scripts.[5] The evaluation was performed using `MultEval` (Clark et al., 2011) with the primary metric of Meteor 1.5 (Denkowski and Lavie, 2014). We also report the results using BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) metrics. The winning submissions are indicated by •. These are the top-scoring submissions and those that are not significantly different (based on Meteor scores) according the approximate randomisation test (with p-value $\leq$ 0.05) provided by `MultEval`. Submissions marked with * are not significantly different from the Baseline according to the same test.

### 4.1 Task 1: English → German

Table 6 shows the results on the Test 2018 dataset with a German target language. The first observation is that the best-performing system, MeMAD_1_FLICKR_DE_MeMAD-OpenNMT-mmod_U, is substantially better than other systems, although it uses unconstrained data. The MeMAD team did not submit a constrained or monomodal submission, so we cannot conclude whether this improvement comes from the use of multimodal data or from the additional parallel data. However, as mentioned in Section 3, the

[5] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/

authors themselves state that the gains mainly come from the additional parallel text data in the monomodal system. The vast majority of systems beat the strong text-only Baseline by a considerable margin. For other teams submitting monomodal and multimodal versions of their systems (e.g. CUNI and LIUMCVC), there does not seem to be a marked difference in automatic metric scores.

We can also observe that the ambiguous word evaluations (LTA) does not lead to the same system ranking as the automatic metrics. While this could stem mainly from the fact that the LTA evaluation is only performed on a small subset of the test cases, we consider that these two automatic evaluations are complementary. General translation quality is measured with the standard metrics (BLEU, METEOR and TER), while the LTA evaluations captures the ability of the system to model complex words which, in many cases, could require the use of the image input to disambiguate them.

### 4.2 Task 1: English → French

Table 7 shows the results for the Test 2018 dataset with French as target language. Once again, the MeMAD_1_FLICKR_FR_MeMAD-OpenNMT-mmod_U system performs significantly better than the other systems.[6] For teams submitting monomodal and multimodal versions of their systems (e.g. CUNI and LIUMCVC), there does not seem to be a marked difference in automatic metric scores. Another interesting observation is that in this case the clearly superior performance of the MeMAD_1_FLICKR_FR_MeMAD-OpenNMT-mmod_U system also shows in the LTA evaluation.

All submissions significantly outperformed the English→French baseline system. For this language pair, the evaluation metrics are in better agreement about the ranking of the submissions, however, the LTA metric is once again less correlated.

### 4.3 Task 1: English → Czech

The Czech language is a new addition to the 2018 evaluation campaign. Table 8 shows the results for the Test 2018 dataset with Czech as target language. A smaller number of teams have submitted systems for this language pair. This is a more complex language pair as demonstrated by the lower automatic scores obtained by the systems. The best results are obtained by the CUNI_1_FLICKR_CS_NeuralMonkeyImagination_U system, under the unconstrained conditions. The constrained systems all perform similarly to each other, and all except CUNI_1_FLICKR_CS_NeuralMonkeyTextual_U are significantly better than the baseline system. Interestingly, for the OSU-BD submissions, LTA seems to disagree significantly with the other metrics. More analysis is necessary to understand why this is the case.

### 4.4 Task 1b: Multisource English, German, French → Czech

Multisource multimodal translation is a new task this year. This task invites participants to use multiple source language inputs, as well as the image, in order to generate Czech translations. Only a few systems have been submitted compared to the other tasks. The results for the Test 2018 dataset are presented in Table 9. We observe that all teams outperformed the text-only baseline, even though in some cases the difference is not significant. No teams used unconstrained data in their submissions.

Again, the LTA results do not follow those of the automatic metrics, particularly for the two top submissions: LTA scores differ by a large margin, while all other metric scores are the same or very similar. This could however result from the very small number of samples available for LTA evaluation for this task: only 52 test instances. Differences in the translation of a very few number of instances can therefore result in considerably differences in LTA scores.

## 5 Human Judgment Results

In addition to the automatic metrics evaluation, we conducted human evaluation to assess the translation quality of the submissions. This evaluation was undertaken for the Task 1 German, French and Czech outputs as well as for the Task 1b Czech outputs for the Test 2018 dataset. This section describes how we collected the human assessments and computed the results. We are grateful to all of the assessors for their contributions.

### 5.1 Methodology

The system outputs indicated as the primary submission were manually evaluated by bilingual Direct Assessment (DA) (Graham et al., 2015) using

[6] We note that their original submission had tokenisation issues, which were fixed by the task organisers.

| **EN → DE** | BLEU ↑ | Meteor ↑ | TER ↓ | LTA ↑ |
|---|---|---|---|---|
| •MeMAD_1_FLICKR_DE_MeMAD-OpenNMT-mmod_U (P) | 38.5 | 56.6 | 44.5 | 47.49 |
| CUNI_1_FLICKR_DE_NeuralMonkeyTextual_U | 32.5 | 52.3 | 50.8 | 46.37 |
| CUNI_1_FLICKR_DE_NeuralMonkeyImagination_U (P) | 32.2 | 51.7 | 51.7 | 47.21 |
| UMONS_1_FLICKR_DE_DeepGru_C (P) | 31.1 | 51.6 | 53.4 | 48.04 |
| LIUMCVC_1_FLICKR_DE_NMTEnsemble_C (P) | 31.1 | 51.5 | 52.6 | 46.65 |
| LIUMCVC_1_FLICKR_DE_MNMTEnsemble_C (P) | 31.4 | 51.4 | 52.1 | 45.81 |
| OSU-BD_1_FLICKR_DE_RLNMT_C (P) | 32.3 | 50.9 | 49.9 | 45.25 |
| OSU-BD_1_FLICKR_DE_RLMIX_C | 32.0 | 50.7 | 49.6 | 46.09 |
| SHEF_1_DE_LT_C | 30.4 | 50.7 | 53.0 | 48.04 |
| SHEF_1_DE_MLT_C (P) | 30.4 | 50.7 | 53.0 | 48.32 |
| SHEF1_1_DE_ENMT_C | 30.8 | 50.7 | 52.4 | 44.41 |
| SHEF1_1_DE_MFS_C (P) | 30.3 | 50.7 | 53.1 | 48.32 |
| LIUMCVC_1_FLICKR_DE_MNMTSingle_C | 28.8 | 49.9 | 55.6 | 45.25 |
| LIUMCVC_1_FLICKR_DE_NMTSingle_C | 29.5 | 49.9 | 54.3 | 47.77 |
| Baseline | 27.6 | 47.4 | 55.2 | 45.25 |
| AFRL-OHIO-STATE_1_FLICKR_DE_4COMBO_U (P) | 24.3 | 45.4 | 58.6 | 46.09 |
| AFRL-OHIO-STATE_1_FLICKR_DE_2IMPROVE_U | 10.0 | 25.4 | 79.0 | 25.42 |

Table 6: Official automatic results for the MMT18 Task 1 on the English → German Test 2018 dataset (ordered by Meteor). Grey background indicate use of resources that fall outside the constraints provided for the shared task. (P) indicate a primary system designated for human evaluation.

| **EN → FR** | BLEU ↑ | Meteor ↑ | TER ↓ | LTA ↑ |
|---|---|---|---|---|
| •MeMAD_1_FLICKR_FR_MeMAD-OpenNMT-mmod_U (P) | 44.1 | 64.3 | 36.9 | 73.08 |
| CUNI_1_FLICKR_FR_NeuralMonkeyTextual_U | 40.6 | 61.0 | 40.7 | 68.44 |
| CUNI_1_FLICKR_FR_NeuralMonkeyImagination_U (P) | 40.4 | 60.7 | 40.7 | 69.29 |
| UMONS_1_FLICKR_FR_DeepGru_C (P) | 39.2 | 60.0 | 41.8 | 68.82 |
| LIUMCVC_1_FLICKR_FR_MNMTEnsemble_C (P) | 39.5 | 59.9 | 41.7 | 68.53 |
| LIUMCVC_1_FLICKR_FR_NMTEnsemble_C (P) | 39.1 | 59.8 | 41.9 | 68.44 |
| SHEF_1_FR_LT_C | 38.8 | 59.8 | 41.5 | 69.57 |
| SHEF_1_FR_MLT_C (P) | 38.9 | 59.8 | 41.5 | 69.86 |
| SHEF1_1_FR_ENMT_C | 38.9 | 59.8 | 41.2 | 67.87 |
| SHEF1_1_FR_MFS_C (P) | 38.8 | 59.7 | 41.6 | 67.58 |
| OSU-BD_1_FLICKR_FR_RLNMT_C (P) | 39.0 | 59.5 | 41.2 | 68.91 |
| OSU-BD_1_FLICKR_FR_RLMIX_C | 38.6 | 59.3 | 41.5 | 67.68 |
| LIUMCVC_1_FLICKR_FR_MNMTSingle_C | 37.9 | 58.5 | 43.4 | 67.77 |
| LIUMCVC_1_FLICKR_FR_NMTSingle_C | 37.6 | 58.4 | 43.2 | 67.11 |
| Baseline | 36.3 | 56.9 | 54.3 | 66.26 |

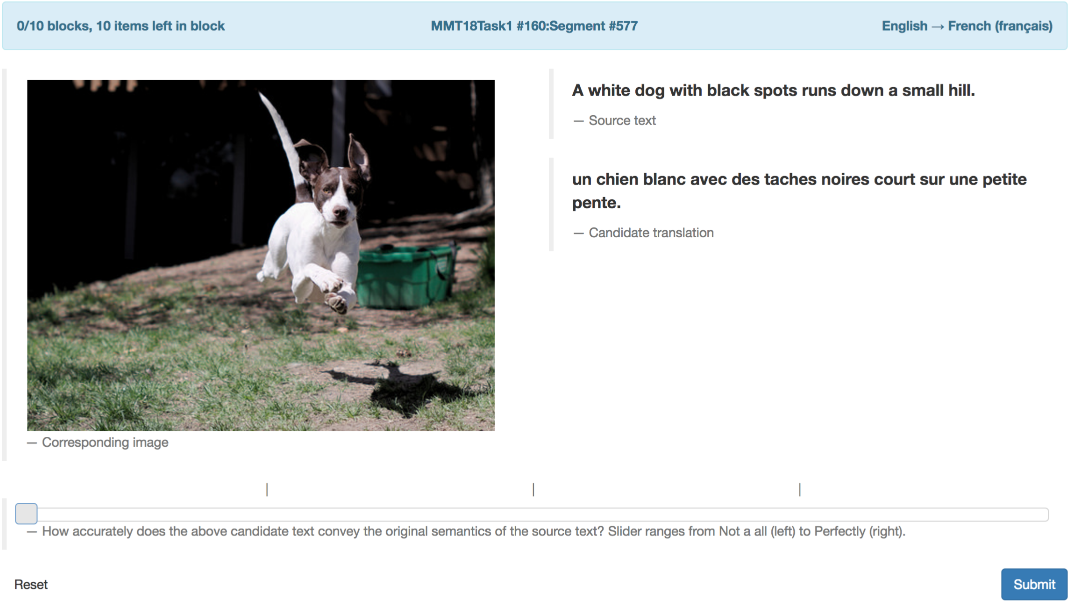
Table 7: Official automatic results for the MMT18 Task 1 on the English → French Test 2018 dataset (ordered by Meteor). Grey background indicate use of resources that fall outside the constraints provided for the shared task. (P) indicate a primary system designated for human evaluation.

| **EN → CS** | BLEU ↑ | Meteor ↑ | TER ↓ | LTA ↑ |
|---|---|---|---|---|
| •CUNI_1_FLICKR_CS_NeuralMonkeyImagination_U (P) | 31.8 | 30.6 | 48.2 | 70.00 |
| OSU-BD_1_FLICKR_CS_RLMIX_C | 30.1 | 29.7 | 51.2 | 54.29 |
| OSU-BD_1_FLICKR_CS_RLNMT_C (P) | 30.2 | 29.5 | 50.7 | 60.71 |
| SHEF1_1_CS_ENMT_C | 29.0 | 29.4 | 51.1 | 71.43 |
| SHEF1_1_CS_MFS_C (P) | 27.8 | 29.2 | 52.4 | 73.57 |
| SHEF_1_CS_LT_C | 28.3 | 29.1 | 51.7 | 72.14 |
| SHEF_1_CS_MLT_C (P) | 28.2 | 29.1 | 51.7 | 71.43 |
| Baseline | 26.5 | 27.7 | 54.4 | 62.14 |
| *CUNI_1_FLICKR_CS_NeuralMonkeyTextual_U | 26.8 | 27.1 | 55.2 | 52.14 |

Table 8: Official automatic results for the MMT18 Task 1 on the English → Czech Test 2018 dataset (ordered by Meteor). Grey background indicate use of resources that fall outside the constraints provided for the shared task. (P) indicate a primary system designated for human evaluation. Submissions marked with * are not significantly different from the Baseline.

| **EN,DE,FR → CS** | BLEU ↑ | Meteor ↑ | TER ↓ | LTA ↑ |
|---|---|---|---|---|
| OSU-BD_1b_CS_RLMIX_C | 26.4 | 28.2 | 52.7 | 55.77 |
| OSU-BD_1b_CS_RLNMT_C (P) | 26.4 | 28.0 | 52.1 | 61.54 |
| SHEF_1b_CS_CON_C | 24.7 | 27.6 | 52.1 | 61.54 |
| *SHEF_1b_CS_MLTC_C (P) | 24.5 | 27.5 | 52.5 | 61.54 |
| SHEF1_1b_CS_ARNN_C (P) | 25.2 | 27.5 | 53.9 | 51.92 |
| *SHEF1_1b_CS_ARF_C | 24.1 | 27.1 | 54.6 | 51.92 |
| Baseline | 23.6 | 26.8 | 54.1 | 53.85 |

Table 9: Official automatic results for the MMT18 Task 1b on the English,German,French → Czech Test 2018 dataset (ordered by Meteor). Submissions marked with * are not significantly different from the Baseline.



0/10 blocks, 10 items left in block

MMT18Task1 #160:Segment #577

English → French (français)

A white dog with black spots runs down a small hill.

— Source text

un chien blanc avec des taches noires court sur une petite pente.

— Candidate translation

— Corresponding image

— How accurately does the above candidate text convey the original semantics of the source text? Slider ranges from Not a all (left) to Perfectly (right).

Figure 3: Example of the human direct assessment evaluation interface.

the Appraise platform (Federmann, 2012). The annotators (mostly researchers) were asked to evaluate the semantic relatedness between the source sentence in English and the target sentence in German, French or Czech. For the Multisource Task (1b), only the English source is presented. For the evaluation task, the image was shown along with the source sentence and the candidate translation. Evaluators were ask to rely on the image when necessary to obtain a better understanding of the source sentence (e.g. in cases where the text was ambiguous). Note that the reference sentence is not displayed during the evaluation to avoid influencing the assessment. Instead, as a control experiment to estimate the quality of the reference sentences (and test the quality of the annotations), we included the references as hypotheses for human evaluation. Figure 3 shows an example of the direct assessment interface used in the evaluation. The score of each translation candidate ranges from 0 (the meaning of the source is not preserved in the target language sentence) to 100 (the meaning of the source is "perfectly" preserved). The overall score of a given system ($z$) corresponds to the mean standardised score of its translations.

### 5.2 Results

For Task 1 English-German translation, we collected 3,422 DAs, resulting in a minimum of 300 and a maximum of 324 direct assessments per system submission, respectively. We collected 2,938 DAs for the English-French translations. This results in a minimum of 280 and a maximum of 307 direct assessments per system submission, respectively. We collected 8,096 DAs for the Task 1 English-Czech translation, representing a minimum of 1,330 and a maximum of 1,370 direct assessments per system submission. For Task 1b English,German,French→Czech translation, we collected 6,827 direct assessments. The least evaluated system received 1,345 assessments, while the most evaluated system received 1,386 direct assessments.

Tables 10, 11, 12 and 13 show the results of the human evaluation for the English to German, English to French and English to Czech Multimodal Translation task (Test 2018 dataset) as well as the Multisource Translation task. The systems are ordered by standardised mean DA scores and clustered according to the Wilcoxon signed-rank test at p-level p $\leq$ 0.05. Systems within a cluster are considered tied. The supplementary Wilcoxon signed-rank scores can be found in Tables 14, 15 and 16 in Appendix A.

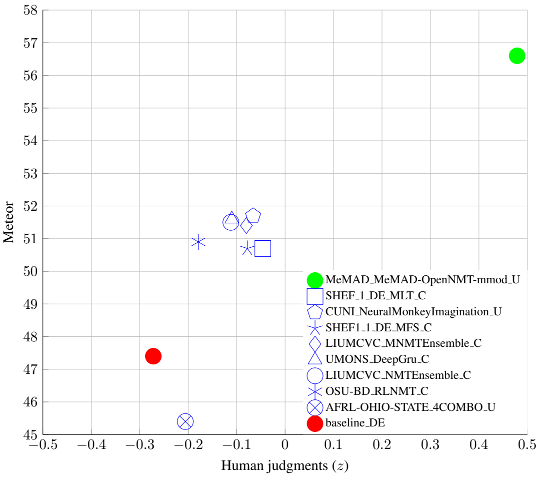


Figure 4: System performance on the English→German Test 2018 dataset as measured by human evaluation against Meteor scores.

The comparison between automatic and human evaluation are presented in Figures 4, 5, 6 and 7. We can observe that METEOR scores are well correlated with the human evaluation.

## 6 Discussion

As mentioned in Section 5, we included the reference sentences in the DA evaluation as if they were candidate translations generated by a system. The first observation is that for all language pairs and all tasks, the references (see gold_* in Tables 10, 11, 12 and 13) are significantly better than all automatic systems with average raw scores above 90%. This does not only validates the references but also the DA evaluation process.

For the first time in the MMT evaluation campaign series, using additional (unconstrained) data resulted in some significant improvement both in terms of automatic score and human evaluation. The biggest improvements come from the unconstrained MeMAD system (for the English-German and English-French), which achieves large improvements in Meteor score compared to the second best system. This is also the case in terms of human evaluation. For English-German, for example, the average raw DA score (87.2, see second column of Table 10) is only 4.5% away from the result of the reference evaluation (91.7). The MeMAD team use a transformer NMT architec-

| | | | **English→German** |
|---|---|---|---|
| # | Ave % | Ave $z$ | System |
| 1 | 91.7 | 0.69 | gold_DE_1 |
| 2 | 87.2 | 0.479 | MeMAD_MeMAD-OpenNMT-mmod_U |
| 3 | 73.5 | -0.046 | SHEF_1_DE_MLT_C |
| | 73.8 | -0.066 | CUNI_NeuralMonkeyImagination_U |
| | 72.6 | -0.078 | SHEF1_1_DE_MFS_C |
| | 71.6 | -0.08 | LIUMCVC_MNMTEnsemble_C |
| | 72.1 | -0.11 | UMONS_DeepGru_C |
| | 72.5 | -0.112 | LIUMCVC_NMTEnsemble_C |
| | 71.1 | -0.179 | OSU-BD_RLNMT_C |
| | 68.6 | -0.206 | AFRL-OHIO-STATE_4COMBO_U |
| | 67.4 | -0.272 | baseline_DE |

Table 10: Results of the human evaluation of the WMT18 English-German Multimodal Translation task (Test 2018 dataset). Systems are ordered by standardized mean DA scores ($z$) and clustered according to Wilcoxon signed-rank test at p-level $p \leq 0.05$. Systems within a cluster are considered tied, although systems within a cluster may be statistically significantly different from each other (see Table 14). Systems using unconstrained data are identified with a gray background.

| | | | **English→French** |
|---|---|---|---|
| # | Ave % | Ave $z$ | System |
| 1 | 90.3 | 0.487 | gold_FR_1 |
| 2 | 86.8 | 0.349 | MeMAD_MeMAD-OpenNMT-mmod_U |
| 3 | 78.5 | 0.047 | CUNI_NeuralMonkeyImagination_U |
| | 77.3 | -0.005 | UMONS_DeepGru_C |
| | 74.9 | -0.05 | LIUMCVC_NMTEnsemble_C |
| | 74.9 | -0.075 | SHEF1_1_FR_MFS_C |
| | 74.5 | -0.088 | SHEF_1_FR_MLT_C |
| | 73.0 | -0.11 | LIUMCVC_MNMTEnsemble_C |
| | 74.4 | -0.12 | OSU-BD_RLNMT_C |
| | 66.0 | -0.376 | baseline_FR |

Table 11: Results of the human evaluation of the WMT18 English-French Multimodal Translation task (Test 2018 dataset). Systems are ordered by standardized mean DA score ($z$) and clustered according to Wilcoxon signed-rank test at p-level $p \leq 0.05$. Systems within a cluster are considered tied, although systems within a cluster may be statistically significantly different from each other (see Table 15). Systems using unconstrained data are identified with a gray background.

| | | | **English→Czech** |
|---|---|---|---|
| # | Ave % | Ave $z$ | System |
| 1 | 93.2 | 0.866 | gold_CS_1 |
| 2 | 70.2 | 0.097 | CUNI_NeuralMonkeyImagination_U.txt |
| | 62.4 | -0.162 | SHEF_1_CS_MLT_C |
| | 60.6 | -0.225 | SHEF1_1_CS_MFS_C |
| | 59.1 | -0.248 | OSU-BD_RLNMT_C |
| 3 | 57.8 | -0.337 | baseline_CS |

Table 12: Results of the human evaluation of the WMT18 English-Czech Multimodal Translation task (Test 2018 dataset). Systems are ordered by standardized mean DA score ($z$) and clustered according to Wilcoxon signed-rank test at p-level p $\leq$ 0.05. Systems within a cluster are considered tied, although systems within a cluster may be statistically significantly different from each other (see Table 16). Systems using unconstrained data are identified with a gray background.

| | | | **English,French,German→Czech** |
|---|---|---|---|
| # | Ave % | Ave $z$ | System |
| | 93.6 | 0.803 | gold_CS_1b |
| | 63.3 | -0.149 | SHEF_1b_CS_MLTC_C |
| | 61.8 | -0.178 | SHEF1_1b_CS_ARNN_C |
| | 62.1 | -0.206 | OSU-BD_1b_CS_RLNMT_C |
| | 59.4 | -0.284 | baseline_CS_task1b |

Table 13: Results of the human evaluation of the WMT18 English,French,German-Czech Multisource Multimodal Translation task (Test 2018 dataset). Systems are ordered by standardized mean DA score ($z$) and clustered according to Wilcoxon signed-rank test at p-level p $\leq$ 0.05. Systems within a cluster are considered tied, although systems within a cluster may be statistically significantly different from each other (see Table 17).

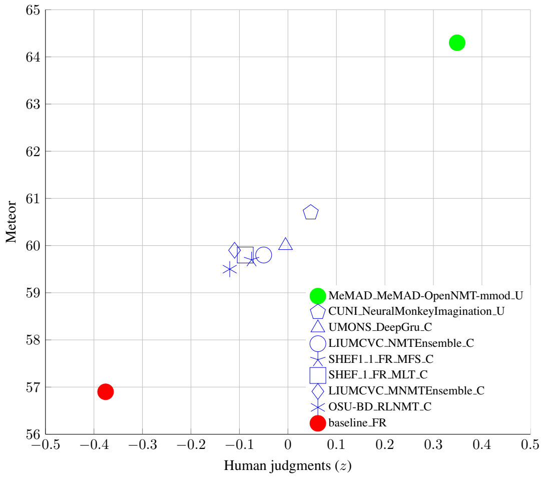


Figure 5: System performance on the English→French Test 2018 dataset as measured by human evaluation against Meteor scores.

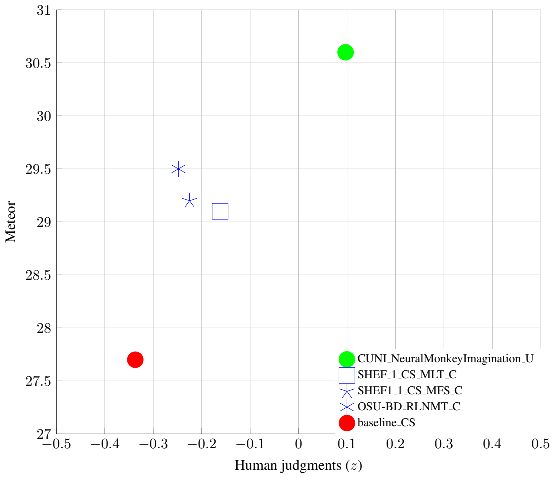


Figure 6: System performance on the English→Czech Test 2018 dataset as measured by human evaluation against Meteor scores.

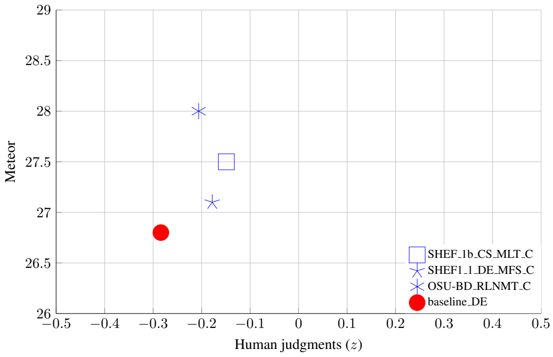


Figure 7: System performance on the English,German,French→Czech Test 2018 dataset as measured by human evaluation against Meteor scores.

ture (as opposed to recurrent neural networks) combined with global image feature that are different from the ResNet features made available by the task organisers. However, according to the authors it seems that most of the improvements come from the additional parallel data.

Many teams proposed a combination of several systems. This is the case for AFRL-OHIO-STATE, LIUMCVC, OSU-BAIDU and SHEF teams. LIUMCVC also submitted a non-ensembled version of each system. Their conclusion is that ensembling multiple systems benefit monomodal and multimodal systems.

**Lexical Translation Accuracy** LTA was a new evaluation for this campaign. Unlike other automatic metrics, LTA only evaluates a specific aspect of translation quality, namely lexical disambiguation. One of the motivations for multimodality in machine translation is that the visual features could help to disambiguate ambiguous words (Elliott et al., 2015; Hitschler et al., 2016). Our aims in introducing the LTA metric was to directly evaluate the disambiguation performance of participating systems.

The LTA columns in Tables 6, 7, 8, and 9 show some interesting trends. First, for teams submitting text-only and multimodal variants of models, the multimodal versions seem to perform better at LTA compared to their text-only counterparts (e.g. CUNI's systems). This trend is not visible using the Meteor, BLEU, or TER metrics. Second, the SHEF systems that were built precisely to perform cross-lingual LTA-style WSD perform well on this metric but they are not always the best-performing system on this metric.

**Multisource multimodal translation** Only two teams participated in this task. The automatic results are presented in Table 9, the human evaluation results are presented in Table 13 and the comparison between automatic and human evaluation results are shown in Figure 6. Although many direct assessments have been collected for this task, it was not possible to separate the systems into different clusters. We can see that there is still a large margin between the performance of the systems and the human gold reference, but this was also the case for the English-Czech language pair in Task 1.

## 7 Conclusions

We presented the results of the third shared task on multimodal translation. The shared task attracted submissions from seven groups, who submitted a total of 45 systems across the two proposed tasks. The Multimodal Translation task attracted the majority of the submissions, with fewer groups attempting multisource multimodal translation.

The main findings of the shared task are:

(i) Additional data can greatly improve the results as demonstrated by the winning unconstrained systems.

(ii) Almost all systems achieved better results compared to the baseline text-only translation system. Various text and visual integration schemes have been proposed, leading to only slight changes in the automatic and human evaluation results.

(iii) Automatic metrics and human evaluation provided similar results. However, it is difficult to evaluate the impact of the multimodality. In the future, submission of monomodal equivalent of the systems will be encouraged in order to better emphasize the effect of using the visual inputs.

We are considering to change the data in favor of a more ambiguous task where all modalities should be used in order to generate the output. A possibility would be to re-use the list of ambiguous words extracted for LTA computation and select the image/sentence pairs containing one or more of those words.

## Acknowledgements

This work was supported by the CHIST-ERA M2CR project (French National Research Agency No. ANR-15-CHR2-0006-01 – Loïc Barrault and Fethi Bougares), by the MultiMT project (EU H2020 ERC Starting Grant No. 678017 – Lucia Specia and Chiraag Lala), and by an Amazon Research Award (Desmond Elliott). We thank the Charles University team for collecting and describing the Czech data. The Czech data collection was supported by the Czech Science Foundation, grant number P103/12/G084.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, pages 1171–1179. MIT Press.

Ozan Caglayan, Adrien Bardet, Fethi Bougares, Loïc Barrault, Kai Wang, Marc Masana, Luis Herranz, and Joost van de Weijer. 2018. LIUM-CVC submissions for WMT18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Ozan Caglayan, Loïc Barrault, and Fethi Bougares. 2016. Multimodal attention for neural machine translation. *CoRR*, abs/1609.03976.

Ozan Caglayan, Mercedes García-Martínez, Adrien Bardet, Walid. Aransa, Fethi. Bougares, and Loïc Barrault. 2017. NMTPY: A Flexible Toolkit for Advanced Neural Machine Translation Systems. *CoRR*, 1706.00457.

Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181.

Jean-Benoit Delbrouck and Stéphane Dupont. 2018. Umons submission for wmt18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *EACL 2014 Workshop on Statistical Machine Translation*.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2013)*, pages 644–649.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 215–233, Copenhagen, Denmark.

Desmond Elliott, Stella Frank, and Eva Hasler. 2015. Multi-language image description with neural sequence models. *CoRR*, abs/1510.04709.

Desmond Elliott, Stella Frank, Khalil Simaan, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *5th Workshop on Vision and Language*, pages 70–74.

Desmond Elliott and Ákos Kádár. 2017. Imagination improves Multimodal Translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 130–141, Taipei, Taiwan.

Christian Federmann. 2012. Appraise: An open-source toolkit for manual evaluation of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. 2018. Detectron. `https://github.com/facebookresearch/detectron`.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2015. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Stig-Arne Grönroos, Benoit Huet, Mikko Kurimo, Jorma Laaksonen, Bernard Merialdo, Phu Pham, Mats Sjöberg, Umut Sulubacak, Jörg Tiedemann, Raphael Troncy, and Raúl Vázquez. 2018. The MeMAD submission to the WMT18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Jeremy Gwinnup, Joshua Sandvick, Michael Hutt, Grant Erdmann, John Duselis, and James Davis. 2018. The afrl-ohio state wmt18 multimodal system: Combining visual with traditional. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Jindřich Helcl, Jindřich Libovický, and Dušan Variš. 2018. CUNI system for the WMT18 multimodal translation tasks. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Julian Hitschler, Shigehiko Schamoni, and Stefan Riezler. 2016. Multimodal Pivots for Image Caption Translation. In *54th Annual Meeting of the Association for Computational Linguistics*, pages 2399–2409.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *45th Annual meeting of Association for Computational Linguistics*, pages 177–180.

Chiraag Lala, Pranava Madhyastha, Carolina Scarton, and Lucia Specia. 2018. Sheffield submissions for wmt18 multimodal translation shared task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Chiraag Lala and Lucia Specia. 2018. Multimodal Lexical Translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1179–1195, Honolulu, Hawaii.

Martin Riedl and Chris Biemann. 2016. Unsupervised compound splitting with distributional semantics rivals supervised methods. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 617–622.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *First Conference on Machine Translation*, pages 543–553.

Jana Straková, Milan Straka, and Jan Hajič. 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. *arXiv preprint arXiv:1603.07012*.

Renjie Zheng, Yilin Yang, Mingbo Ma, and Liang Huang. 2018. Ensemble sequence level training for multimodal mt: Osu-baidu wmt18 multimodal machine translation system report. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

## A Significance tests

Tables 14, 15 and 16 show the Wilcoxon signed-rank test used to create the clustering of the systems.

**English→German**

| | gold_DE_1 | MeMAD_MeMAD-OpenNMT-mmod_U | SHEF_1_DE_MLT_C | CUNI_NeuralMonkeyImagination_U | SHEF1_1_DE_MFS_C | LIUMCVC_MNMTEnsemble_C | UMONS_DeepGru_C | LIUMCVC_NMTEnsemble_C | OSU-BD_RLNMT_C | AFRL-OHIO-STATE_4COMBO_U | baseline_DE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| gold_DE_1 | - | 9.3e-10 | 1.2e-30 | 2.1e-28 | 2.0e-30 | 8.1e-28 | 1.8e-29 | 5.9e-31 | 2.5e-36 | 3.7e-37 | 4.2e-38 |
| MeMAD_MeMAD-OpenNMT-mmod_U | - | - | 4.4e-14 | 1.5e-12 | 2.7e-14 | 1.0e-12 | 7.0e-14 | 5.5e-15 | 1.2e-19 | 3.2e-21 | 3.5e-22 |
| SHEF_1_DE_MLT_C | - | - | - | - | - | - | - | - | 5.0e-02 | 1.2e-02 | 3.0e-03 |
| CUNI_NeuralMonkeyImagination_U | - | - | - | - | - | - | - | - | 4.8e-02 | 1.2e-02 | 3.1e-03 |
| SHEF1_1_DE_MFS_C | - | - | - | - | - | - | - | - | - | 2.9e-02 | 8.3e-03 |
| LIUMCVC_MNMTEnsemble_C | - | - | - | - | - | - | - | - | - | 3.2e-02 | 8.8e-03 |
| UMONS_DeepGru_C | - | - | - | - | - | - | - | - | - | - | 1.5e-02 |
| LIUMCVC_NMTEnsemble_C | - | - | - | - | - | - | - | - | - | - | 2.0e-02 |
| OSU-BD_RLNMT_C | - | - | - | - | - | - | - | - | - | - | - |
| AFRL-OHIO-STATE_4COMBO_U | - | - | - | - | - | - | - | - | - | - | - |
| baseline_DE | - | - | - | - | - | - | - | - | - | - | - |

Table 14: English → German Wilcoxon signed-rank test at p-level $p \leq 0.05$. ‘-’ means that the value is higher than 0.05.

| English→French | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | gold_FR_1 | MeMAD_MeMAD-OpenNMT-mmod_U | CUNI_NeuralMonkeyImagination_U | UMONS_DeepGru_C | LIUMCVC_NMTEnsemble_C | SHEF1_1_FR_MFS_C | SHEF_1_FR_MLT_C | LIUMCVC_MNMTEnsemble_C | OSU-BD_RLNMT_C | baseline_FR |
| gold_FR_1 | - | 2.7e-02 | 1.3e-09 | 1.8e-10 | 1.5e-10 | 5.1e-12 | 6.2e-13 | 4.4e-11 | 6.6e-14 | 3.3e-20 |
| MeMAD_MeMAD-OpenNMT-mmod_U | - | - | 3.0e-05 | 6.6e-06 | 3.0e-06 | 3.3e-07 | 8.1e-08 | 9.0e-07 | 1.4e-08 | 3.9e-14 |
| CUNI_NeuralMonkeyImagination_U | - | - | - | - | - | - | - | - | 4.9e-02 | 5.1e-05 |
| UMONS_DeepGru_C | - | - | - | - | - | - | - | - | - | 2.1e-04 |
| LIUMCVC_NMTEnsemble_C | - | - | - | - | - | - | - | - | - | 6.6e-04 |
| SHEF1_1_FR_MFS_C | - | - | - | - | - | - | - | - | - | 1.9e-03 |
| SHEF_1_FR_MLT_C | - | - | - | - | - | - | - | - | - | 3.5e-03 |
| LIUMCVC_MNMTEnsemble_C | - | - | - | - | - | - | - | - | - | 3.0e-03 |
| OSU-BD_RLNMT_C | - | - | - | - | - | - | - | - | - | 1.0e-02 |
| baseline_FR | - | - | - | - | - | - | - | - | - | - |

Table 15: English → French Wilcoxon signed-rank test at p-level $p \leq 0.05$. '-' means that the value is higher than 0.05.

| | English→Czech | | | | | |
|---|---|---|---|---|---|---|
| | gold_CS_1 | CUNI_NeuralMonkeyImagination_U | SHEF_1_CS_MLT_C | SHEF1_1_CS_MFS_C | OSU-BD_RLNMT_C | baseline_CS_encs |
| gold_CS_1 | - | 6.9e-100 | 5.9e-150 | 3.6e-166 | 8.3e-158 | 1.3e-170 |
| CUNI_NeuralMonkeyImagination_U | - | - | 1.5e-10 | 1.4e-15 | 2.1e-16 | 1.5e-22 |
| SHEF_1_CS_MLT_C | - | - | - | - | 2.2e-02 | 6.7e-05 |
| SHEF1_1_CS_MFS_C | - | - | - | - | - | 8.2e-03 |
| OSU-BD_RLNMT_C | - | - | - | - | - | 2.8e-02 |
| baseline_CS_encs | - | - | - | - | - | - |

Table 16: English → Czech Wilcoxon signed-rank test at p-level $p \leq 0.05$. ‘-’ means that the value is higher than 0.05.

**English,French,German→Czech**

| | gold_CS_1b | SHEF_1b_CS_MLTC_C | SHEF1_1b_CS_ARNN_C | OSU-BD_1b_CS_RLNMT_C | baseline_CS_.encs.task1b |
|---|---|---|---|---|---|
| gold_CS_task1b | - | 4.4e-127 | 1.3e-115 | 3.8e-116 | 4.1e-132 |
| SHEF_1b_CS_MLTC_C | - | - | - | - | 4.3e-03 |
| SHEF1_1b_CS_ARNN_C | - | - | - | - | 1.2e-02 |
| OSU-BD_1b_CS_RLNMT_C | - | - | - | - | - |
| baseline_CS_task1b | - | - | - | - | - |

Table 17: English,French,German → Czech Wilcoxon signed-rank test at p-level p $\leq$ 0.05. '-' means that the value is higher than 0.05.

# Findings of the WMT 2018 Biomedical Translation Shared Task: Evaluation on Medline test sets

**Mariana Neves**
German Federal Institute for
Risk Assessment (BfR),
Germany

**Antonio Jimeno Yepes**
IBM Research,
Australia

**Aurélie Névéol**
LIMSI, CNRS,
Uni. Paris Saclay, France

**Cristian Grozea**
Fraunhofer Institute
FOKUS, Germany

**Amy Siu**
Beuth University of
Applied Sciences, Germany

**Madeleine Kittner**
Humboldt-Universität
zu Berlin, Germany

**Karin Verspoor**
University of Melbourne,
Australia

## Abstract

Machine translation enables the automatic translation of textual documents between languages and can facilitate access to information only available in a given language for non-speakers of this language, e.g. research results presented in scientific publications. In this paper, we provide an overview of the Biomedical Translation shared task in the Workshop on Machine Translation (WMT) 2018, which specifically examined the performance of machine translation systems for biomedical texts. This year, we provided test sets of scientific publications from two sources (EDP and Medline) and for six language pairs (English with each of Chinese, French, German, Portuguese, Romanian and Spanish). We describe the development of the various test sets, the submissions that we received and the evaluations that we carried out. We obtained a total of 39 runs from six teams and some of this year's BLEU scores were somewhat higher that last year's, especially for teams that made use of biomedical resources or state-of-the-art MT algorithms (e.g. Transformer). Finally, our manual evaluation scored automatic translations higher than the reference translations for German and Spanish.

## 1 Introduction

Automatic translation of documents from one language to another facilitates broader information access for resources only available in a particular language. Even in the scientific literature, in which most important articles are published only in English, an increasing number of researchers support citing articles published in other languages for the sake of not missing important research or to avoid carrying out duplicate experiments (Lazarev and Nazarovets, 2018). Recent discussions on this topic in the journal *Nature* have appealed for translation of the best Chinese papers (Tao et al., 2018) and the development of automatic tools for the automatic translation of publications (Prieto, 2018).

Therefore, biomedicine is a domain for which suitable parallel corpora, official evaluation test sets and machine translation (MT) systems are in high demand. There is active development of parallel corpora in this domain (see the recent survey in (Névéol et al., 2018)). In this year alone, three new corpora have been published in a single conference: a compilation of full texts from the Scielo database for English, Portuguese, and Spanish (Soares et al., 2018), medical documents and glossaries for Spanish/English (Villegas et al., 2018) and a biomedical corpus for Romanian (Mitrofan and Tufis, 2018). However, in spite of the growing number of parallel corpora and the many open source tools for MT (e.g., Moses (Koehn et al., 2007), OpenNMT (Klein et al., 2017) and Marian (Junczys-Dowmunt et al., 2018)), there is still no ready-to-use tool for automatic translation of biomedical publications for any language pair.

With the aim of fostering advances in this field, we organized the third edition of the Biomedical Translation Task in the Conference for Machine Translation (WMT).[1] It builds on the two previous editions (Bojar et al., 2016; Jimeno Yepes et al., 2017) by offering test sets from Medline for six

[1] http://www.statmt.org/wmt18/biomedical-translation-task.html

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 324–339
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64030

language pairs and from EDP for one language pair, as detailed below:

- Chinese-English (zh/en); Eng.-Chinese (en/zh)
- French-English (fr/en); Eng.-French (en/fr)
- German-English (de/en); Eng.-German (en/de)
- Portuguese-English (pt/en); Eng.-Port. (en/pt)
- English-Romanian (en/ro)
- Spanish-English (es/en); Eng.-Span. (en/es)

Most test sets were derived from scientific abstracts from Medline which were available in both languages. Except for Romanian, we addressed translation in both directions for all language pairs. This was not possible for Romanian due to the low number (less than 50) of parallel abstracts which are available in Medline. For the first time, we have an Asian language, specifically Chinese.

In this paper, we describe details of the challenge. Section 2 presents the construction and quality analysis of the test sets, followed by the details on the six participating teams in Section 3. Section 4 presents the results for both automatic and manual evaluation that we carried out, as well as some additional evaluations which are new this year. Finally, we provide a comprehensive discussion of the results and quality of the translations in Section 5.

## 2 Test sets

Test sets were obtained from Medline and EDP. In these sources, text for both languages is readily available from the authors of the publications.

**EDP.** This year's test set was derived from last year's processing of publications. We kept one extra test set for this year's challenge. It can be noted that the sentence segmentation offered for the EDP corpus this year was performed manually. More details can be found in the description of the challenge in 2017 (Jimeno Yepes et al., 2017).

**MEDLINE.** We constructed the various Medline test sets following a similar strategy carried out for the Scielo corpus (Neves et al., 2016). We started by downloading MEDLINE 2018 and retrieving those entries whose abstract was available for more than one language, usually English was one of the languages. Such abstracts are identified by the XML tag *OtherAbstract* and its attribute *Language*. We only considered the abstract of the publications since the titles were frequently only available in one language. We randomly selected a subset of the abstracts for the six language pairs under consideration and for which we have native speakers of the foreign languages.

The text of the abstracts were extracted from the XML files and 120 abstracts were randomly selected, excepted for Romanian whose total of parallel documents in Medline was less than 50. The number 120 accounts for possible errors in the preprocessing of the abstract in order to have a final test set of 100 abstracts to be split into the two translation directions. The documents were automatically split using the Stanford CoreNLP tool and the respective available models for each languages, i.e., Chinese, French, German and Spanish (Manning et al., 2014).[2] Since for Portuguese and Romanian no models are available in the Stanford CoreNLP tools, we used models for other similar Roman languages (Spanish for Portuguese and French for Romanian). The sentences were then automatically aligned using the GMA tool for which we provided a list of stopwords for each language.[3] After a short analysis of the alignment of the Chinese/English abstracts, and given the bad alignments that we obtained, we carried out a new automatic alignment using the Champollion tool (Ma, 2006).[4] The resulting aligned sentences were then manually checked for assessing their quality.

### 2.1 Manual evaluation of the automatic alignment

After compiling the Medline test sets, we manually checked the totality of the abstracts to assess the quality of the automatic alignment (cf. results shown in Table 2). We utilized a modified version of the Quality Checking task of our installation of the Appraise tool (Federmann, 2010, 2018) and one native speaker of each non-English language carried out the validation (cf. Figure 1). The only exception were the Chinese abstracts which were manually checked without the use of the Appraise tool. For each language pair, we checked the totality of the abstracts for both translation directions, e.g., en/de and de/en, which was later randomly

[2] https://stanfordnlp.github.io/CoreNLP/

[3] https://nlp.cs.nyu.edu/GMA/

[4] http://champollion.sourceforge.net/

| Test sets | | de/en | fr/en | pt/en | es/en | en/de | en/fr | en/pt | en/es | en/ro | en/zh | zh/en |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EDP | # documents | | 86 | | | | 83 | | | | | |
| | # sentences | | 879/880 | | | | 823/821 | | | | | |
| Medline | # documents | 48 | 49 | 50 | 50 | 48 | 49 | 50 | 50 | 40 | 50 | 49 |
| | # sentences | 342/337 | 318/328 | 283/286 | 286/300 | 352/378 | 279/281 | 332/318 | 299/263 | 301/293 | 311/307 | 279/311 |

Table 1: Overview of the test sets. We present the number of documents and sentences in each test set. The number of sentences might be different for the two languages in a test set.

split into two test sets. The only exception was the Romanian test set. Due to its small size, we only built one test set for one translation direction (en/ro).

The number of completely unaligned sentences was rather uniform across the various language pairs and usually less than 5%, with the exception of Spanish (more than 8%) and German (more that 15%). All other partial alignments (Overlap, Source>Target and Target>Source) had a contribution of less than 10%. For three languages, at least 80% of the sentences were correctly aligned, while for Spanish and German only 70% and 65% of the sentences were correctly aligned. The lower quality of these two test sets could certainly affect the calculation of the BLEU score and we will address this problem later in Section 4.2.3.

During the manual validation, we detected problems in the parallel abstracts. For instance, four abstracts (PMIDs 24616752, 25767637, 26941877 and 24294348) in the German test set had to be excluded because they were wrongly tagged in Medline as being in German, while they were written in Italian. The same occurred in the French test sets, were two abstracts (PMIDs 23396711 and 24883131) were also in Italian. This suggests that the use of automatic language identification tools could be useful to validate the language metadata retrieved from MEDLINE.

## 3 Participating teams and systems

We received submissions from six teams, as summarized in Table 3. The teams came from research and academic institutions of four countries (Brazil, Germany, Spain and USA) and from three continents. An overview of the teams and their systems is provided below.

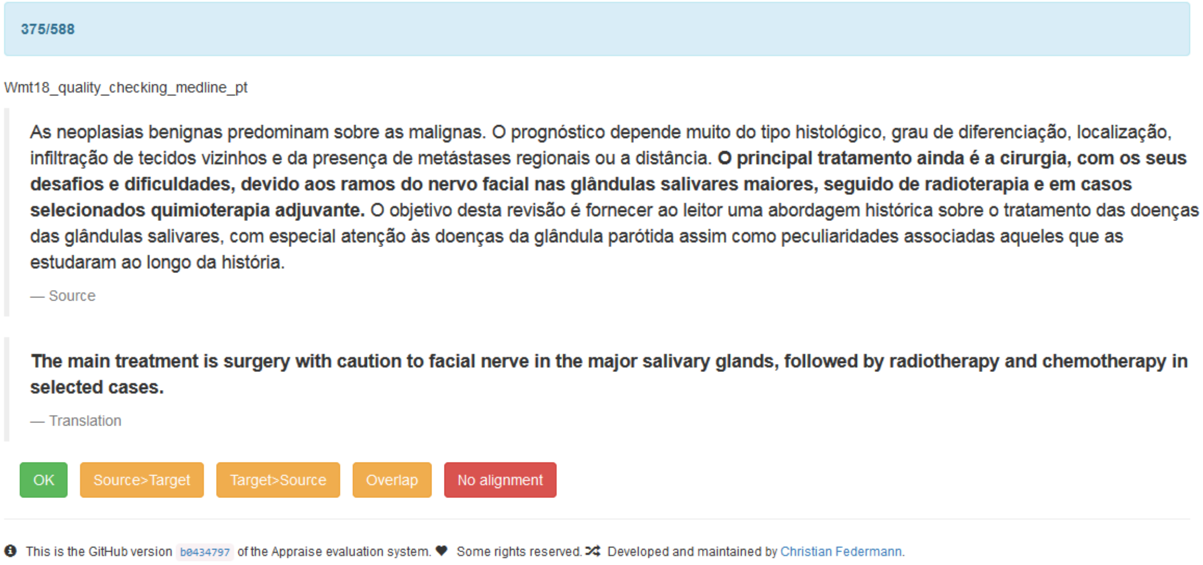
**FOKUS (Fraunhofer Institute FOKUS, Germany).** The FOKUS team participated with a system based on neural machine translation (NMT) based on the implementation of the Transformer architecture (Kaiser et al., 2017; Vaswani et al., 2017) for MT (Grozea, 2018). The NMT system made use of biomedical and news corpora for either training or validation (tuning). In addition to this, and in order to automatically select the highest fidelity translation, they developed heuristics based on a dictionary and on stemming. Further, they performed diacritics normalization in order to account for recent ortographic changes in the Romanian language.

**Hunter MT (Hunter College, USA).** The Hunter team (Khan et al., 2018) used different transfer learning methods and trained different in-domain biomedical data sets one after another. Their system was set up using parameters of previous training as the initialization of the following training. A News based model was used as pre-training.

**LMU (Ludwig Maximilian University of Munich, Germany).** The LMU team implementated various neural network models and trained and tuned the models on parallel biomedical data (Huck et al., 2018). They experimented with implementations of the Transformer architecture (Sockeye implementation) and the encoder-decoder models (Nematus toolkit). The authors highlight that the word segmentation used on the German language for both translation directions were responsible for the good performance of the system in the human evaluation.

**TFG TALP UPC (Technical University of Catalunya, Spain).** For their system that provides translations into English, the TGF TALP UPC team participated with a Transformer architecture (Kaiser et al., 2017; Vaswani et al., 2017) using both single-language and multi-source systems (Tubay and Costa-Jussà, 2018). The systems were trained on the Scielo and Medline titles made available by the shared task in the last years. The multi-source systems utilized a concatenation of training data from es/en, fr/en and pt/en.

**UFRGS (Universidade Federal do Rio Grande do Sul, Brazil).** The UFRGS team participated with two runs based either on Moses (Koehn et al.,



As neoplasias benignas predominam sobre as malignas. O prognóstico depende muito do tipo histológico, grau de diferenciação, localização, infiltração de tecidos vizinhos e da presença de metástases regionais ou a distância. **O principal tratamento ainda é a cirurgia, com os seus desafios e dificuldades, devido aos ramos do nervo facial nas glândulas salivares maiores, seguido de radioterapia e em casos selecionados quimioterapia adjuvante.** O objetivo desta revisão é fornecer ao leitor uma abordagem histórica sobre o tratamento das doenças das glândulas salivares, com especial atenção às doenças da glândula parótida assim como peculiaridades associadas aqueles que as estudaram ao longo da história.

— Source

**The main treatment is surgery with caution to facial nerve in the major salivary glands, followed by radiotherapy and chemotherapy in selected cases.**

— Translation

This is the GitHub version b0434797 of the Appraise evaluation system. ♥ Some rights reserved. Developed and maintained by Christian Federmann.

Figure 1: Screen-shot of Appraise during manual validation of the dataset for Portuguese.

| Test sets | No alignment | OK | Overlap | Source > Target | Target > Source | Total |
|---|---|---|---|---|---|---|
| en/de, de/en | 104 (15.38%) | 437 (64.64%) | 23 (3.40%) | 60 (8.88%) | 52 (7.69%) | 676 |
| en/es, es/en | 46 (8.30%) | 388 (70.04%) | 38 (6.86%) | 44 (7.94%) | 38 (6.86%) | 554 |
| en/fr, fr/en | 20 (3.36%) | 528 (88.59%) | 6 (1.01%) | 20 (3.36%) | 22 (3.69%) | 596 |
| en/pt, pt/en | 11 (1.87%) | 490 (83.33%) | 9 (1.53%) | 41 (6.97%) | 37 (6.29%) | 588 |
| en/ro | 7 (2.39%) | 260 (88.74%) | 3 (1.02%) | 12 (4.10%) | 11 (3.75%) | 293 |
| en/zh, zh/en | 19 (3.26%) | 528 (90.72%) | 4 (0.69%) | 18 (3.09%) | 13 (2.23%) | 582 |

Table 2: Manual validation of the automatic alignment sentences for the Medline test sets. Values are shown in absolute and percentage numbers. The test include the abstracts for both languages directions, with the exception of the Romanian language. The total column represents the totality of the aligned sentences

| Team ID | Institution |
|---|---|
| FOKUS | Fraunhofer Institute FOKUS (Germany) |
| Hunter MT | Hunter College (USA) |
| LMU | Ludwig Maximilian University of Munich (Germany) |
| TFG TALP UPC | Technical University of Catalunya (Spain) |
| UFRGS | Universidade Federal do Rio Grande do Sul (Brazil) |
| UHH-DS | University of Hamburg (Germany) |

Table 3: List of the participating teams.

2007) or OpenNMT (Klein et al., 2017) systems (Soares and Becker, 2018). Training data was prepared by concatenating several in-domain and out-of-domain resources. The in-domain corpora included scientific articles (full texts) from Scielo, the UFAL medical corpus, the EMEA corpus and Brazilian theses and dissertations. Due to possible overlap with the test sets from Medline, the team applied some procedures to automatically exclude some publications from the Scielo training data. Terminological resources such as the Unified Medical Language System (UMLS) (Bodenreider, 2004) were used as well.

**UHH-DS (University of Hamburg, Germany).** The UHH-DS team utilized Moses (Koehn et al., 2007) trained on a variety of in-domain and general domain corpora (Duma and Menzel, 2018). The main feature of their system was the development of an unsupervised method to automatically under-sample sentences from the general domain collection that were better suited for the biomedical domain. Their under-sampling algorithm can be applied either on the source or target side of the corpora, as well as on both sides.

# 4 Evaluation

In this section we describe the various submissions that we obtained and present the results that these achieved based on both automatic and manual valuation.

## 4.1 Submissions

In total, we received 39 submissions from the six teams, as summarized in Table 4. Unfortunately, we received no submissions for Chinese (neither zh/en nor en/zh) and no submissions for the French EDP test set (fr/en).

**FOKUS.** The FOKUS team submitted two runs in which one (run1) was trained on a biomedical corpus and validated on news corpora while the second one (run2, primary run) is an ensemble of various NMT systems and uses the heuristics they defined for selecting the best translation.

**Hunter.** The Hunter's team submitted two runs for en/fr for each of the Medline and EDP test sets. In these runs, they considered NMT based ensembles and trained on various in-domain and out-of-the-domain corpora. However, differences between the runs are unclear.

**LMU.** The three en/de submissions from the LMU team were the following: a right-to-left reranked Transformer (run1, primary run), a Transformer ensemble without re-ranking (run2) and the encoder-decoder built with Nematus (run3). The only submission for de/en was a Transformer without ensemble.

**TFG TALP UPC.** Each two submissions for language pairs es/en, fr/en and pt/en utilized either multi-source (run1, primary run) or the single-source (run2) training.

**UFRGS.** The two submissions from the UFRGS teams seem to have differed only on the MT tool that they used, i.e., either OpenNMT (run1, primary run) or Moses (run2).

**UHH-DS.** The three submissions for each of the language pairs (en/es, en/pt, en/ro, es/en and pt/en) differed on whether the under-sampling algorithm was applied only on the English side (run1), on the non-English side (run2) or on both sides (run3, primary run).

## 4.2 Automatic evaluation

Here we provide the results for the automatic evaluation and rank the systems regarding the resulting scores. We computed BLEU scores at the sentence level using the script `mteval-v14.pl` from the Moses distribution.[5] For all test sets and language pairs, we compare the submissions (automatic translations) to the respective reference one.

### 4.2.1 Automatic evaluation: EDP test sets

The BLEU scores for the EDP test set are presented in Table 5. Given that we received only two submissions from a single team, we could not perform comparison between teams. We ranked the two submissions as follows:

- en/fr: Hunter (run 1) $<$ Hunter (run 2).

Run2 obtained a slightly higher score than run1, however, reasons for this improvement are unknown.

### 4.2.2 Automatic evaluation: Medline test sets

This year, we calculated BLEU scores based on the totality of the sentences (including the ones with incorrect alignments) as well as based only

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

| Teams | de/en | en/de | en/es | en/fr | en/pt | en/ro | es/en | fr/en | pt/en | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| FOKUS | | | | | | M2 | | | | 2 |
| Hunter | | | | E2M2 | | | | | | 4 |
| LMU | M | M3 | | | | | | | | 4 |
| TFG TALP UPC | | | | | | | M2 | M2 | M2 | 6 |
| UFRGS | | | M2 | | M2 | | M2 | | M2 | 8 |
| UHH-DS | | | M3 | | M3 | M3 | M3 | | M3 | 15 |
| Total | 1 | 3 | 5 | 4 | 5 | 5 | 7 | 2 | 7 | 39 |

Table 4: Overview of submissions for each language pair and test set: [M]edline and [E]DP. The number next to the letter indicates the number of runs that the team submitted for the corresponding test set (if larger than one).

| Team | Runs | en/fr |
|---|---|---|
| Hunter | run1 | 22.20 |
| | run2 | 23.24* |

Table 5: BLEU scores for the EDP en/fr dataset. * indicates the primary run as informed by the participants.

on the sentences which were perfectly aligned (cf. Section 2).

BLEU scores for the Medline test set are presented in Table 6. For some language pairs, i.e., de/en, en/de, en/fr and fr/en, we could not compare results between various teams since we received submissions only from one team. Moreover, we only received one submission from one team for de/en. Therefore, no further comparison was possible for this language pair. We ranked the various teams and submissions, for those languages for which we received more than one submission, as follows:

- en/de: LMU (run3) < LMU (runs 1,2);
- en/es: UHH-DS (runs 1,2,3) < UFRGS (runs 1,2);
- en/fr: Hunter (runs 1,2);
- en/pt: UHH-DS (runs 1,2,3) < UFRGS (runs 1,2);
- en/ro: UHH-DS (runs 1,2,3) < FOKUS (run 1) < FOKUS (run 2);
- es/en: UHH-DS (run 2) < UHH-DS (runs 1,3) < TGF TALP UPC (runs 1,2) < UFRGS (runs 1,2);
- fr/en: TGF TALP UPC (run 2) < TGF TALP UPC (run 1);
- pt/en: TGF TALP UPC (run 2) < TGF TALP UPC (run 1) < UHH-DS (runs 1,2,3) < UFRGS (runs 1,2).

In the following we provide a short summary of the results with regard to the method or resources that have been used.

**de/en.** The run based on the Transformer architecture from the LMU team obtained a reasonable BLEU score. However, we could not compare this to any other submission.

**en/de.** There was little difference in the BLEU score between the two first submissions, both based on the Transformer architecture, but both did seem to be superior to the third run based on the encoder-decoder model.

**en/es.** The best results for en/es were obtained by the UFRGS team when using the Moses system (run2) instead of neural MT (run1), as expected by the team. However, the difference between both submissions is not significant. We observed no significant difference between the three submissions from the UHH-DS team. However, all of them yield much lower BLEU scores than the submissions by the UFRGS team.

**en/fr.** The submissions from the Hunter team obtained very similar scores for the Medline test sets. Details on each run is unclear but these differences seem to have brought significant improvement on the scores only on the EDP test set (cf. Table 5).

**en/pt.** Both submissions from the UFRGS team obtained the highest BLEU scores, which again and similar to the results obtained for en/es, did not confirm the superiority of neural MT. The three runs from the UHH-DS team were closer to the ones from the UFRGS team (in comparison to the ones for en/es), but still rather inferior. This time, run1 (under-sampling based on the English side) did perform a little better than the other two runs, specially regarding run2 (under-sampling based on the non-English side).

| Teams | Runs | de/en | en/de | en/es | en/fr | en/pt | en/ro | es/en | fr/en | pt/en |
|---|---|---|---|---|---|---|---|---|---|---|
| FOKUS | run1 | | | | | | 16.97 | | | |
| | run2 | | | | | | 18.10* | | | |
| Hunter MT | run1 | | | | 23.41 | | | | | |
| | run2 | | | | 23.24* | | | | | |
| LMU | run1 | 23.93* | 18.81* | | | | | | | |
| | run2 | | 18.75 | | | | | | | |
| | run3 | | 17.16 | | | | | | | |
| TFG TALP UPC | run1 | | | | | | | 40.49* | 25.78* | 39.49* |
| | run2 | | | | | | | 39.06 | 19.42 | 38.54 |
| UFRGS | run1 | | | 39.62* | | 39.43* | | 43.31* | | 42.58* |
| | run2 | | | 39.77 | | 39.43 | | 43.41 | | 42.58 |
| UHH-DS | run1 | | | 31.32 | | 34.92 | 14.60 | 36.16 | | 41.84 |
| | run2 | | | 31.05 | | 34.19 | 14.39 | 35.17 | | 41.80 |
| | run3 | | | 31.33* | | 34.49* | 14.07* | 36.05* | | 41.79* |

Table 6: Results for the Medline dataset. * indicates the primary run as informed by the participants.

**en/ro.** The run2 of the FOKUS team, which consisted on an ensemble of various NMT systems and used heuristics for selecting the best translation, obtained the highest BLEU score. The two submissions from UHH-DS reached BLEU scores which were slightly below results from the FOKUS team. We observed no significant difference between the three runs. However, similar to en/pt, under-sampling based on the English side (run1) seems to perform slightly better than under-sampling based on both sides (run3).

**es/en.** Once again the statistical MT system (Moses) from the UFRGS team obtained a slightly higher score than their neural MT system (OpenNMT). Indeed, the BLEU scores obtained by run2 of the team was the highest one among all submissions to the shared task for all language pairs. The following two best scores belonged to the Transformer-based MT systems from the TGF TALP UPC team. Even though a more recent and currently state-of-the-art method (Transformer) was used by team TGF TALP UPC, the better results obtained by UFRGS were probably due to the larger training collection that they used. The model trained on various sources obtained a slightly better score. The three runs from the UHH-DS team were rather inferior than the ones from the two other teams. A significant difference was only observed for run2 (under-sampling based on the non-English side) which achieved lower BLEU scores for this language pair.

**fr/en.** The only submissions for fr/en belonged to the Transformer-based MT systems from the TGF TALP UPC team. This time, the improvement of the multi-source model over the single model was very significant. However, the highest score was rather low in comparison to the other submissions of the team.

**pt/en.** There was no difference in the two submissions from the UFRGS team, both obtained the highest BLEU scores. The three runs from the UHH-DS team obtained the second best results but we observed no significant difference between the three of them. Finally, the two lowest scores were obtained by the Transformer-based MT systems from the TGF TALP UPC team. Similar to results for es/en and fr/en, the system trained on various sources obtained a little improvement over the single models. Also similar to the es/en results, the higher performance from UFRGS was probably due to the use of more resources for training the MT systems.

### 4.2.3 Evaluation for sentences with good alignment

This year, we also calculated additional BLEU scores when considering only the sentences whose alignments we manually classified as being correct. Correctly aligned means that sentences in both languages contained exactly the same information and neither of the sentences contained more information than the other (cf. Section 2).

Results for this subset of the test set are presented in Table 7. For most teams, improvements were significant, ranging from two to four BLEU points, but was up to one point for en2ro. The overall order of the results mostly remained the same.

### 4.2.4 Evaluation for Romanian after diacritics normalization

In the particular case of the Romanian language, there were fairly recent changes in the ortho-

| Teams | Runs | de/en | en/de | en/es | en/fr | en/pt | en/ro | es/en | fr/en | pt/en |
|---|---|---|---|---|---|---|---|---|---|---|
| FOKUS | run1 | | | | | | 17.84 | | | |
| | run2 | | | | | | 19.11* | | | |
| Hunter | run1 | | | | 24.66 | | | | | |
| | run2 | | | | 24.76* | | | | | |
| LMU | run1 | 28.84* | 24.30* | | | | | | | |
| | run2 | | 23.88 | | | | | | | |
| | run3 | | 21.84 | | | | | | | |
| TFG TALP UPC | run1 | | | | | | | 42.91* | 27.10* | 42.55* |
| | run2 | | | | | | | 41.26 | 20.20 | 41.56 |
| UFRGS | run1 | | | 44.50* | | 43.14* | | 46.92* | | 46.01* |
| | run2 | | | 44.50 | | 43.14 | | 46.92 | | 46.01 |
| UHH-DS | run1 | | | 34.77 | | 37.24 | 15.85 | 38.45 | | 44.28 |
| | run2 | | | 34.70 | | 36.76 | 15.62 | 37.17 | | 44.32 |
| | run3 | | | 35.08* | | 36.91* | 15.28* | 38.18* | | 44.27* |

Table 7: Results for the Medline dataset using OK aligned sentences. * indicates the primary run as informed by the participants.

| Teams | Runs | en/ro | en/ro-OK |
|---|---|---|---|
| FOKUS | run1 | 22.17 | 22.98 |
| | run2 | 23.42* | 24.22* |
| UHH-DS | run1 | 15.40 | 15.95 |
| | run2 | 15.09 | 15.69 |
| | run3 | 14.77 | 15.44 |

Table 8: Results for the Medline en2ro test set after normalization of diacritics. * indicates the primary run as informed by the participants.

graphic recommendation with respect to diacritics notation[6]: comma-below ș and Ș should be used instead of cedilla-below ş and Ş; in the same way, the comma-below ț and Ț should be used instead of the cedilla-below ţ and Ţ, according to a 2003 communicate from the "Iorgu Iordan" Institute of Linguistics of the Romanian Academy.

While the two comma-below and cedilla-below variants of those letters are hardly distinguishable to a human reader, they have different unicode codes and thus replacing one with another in a word makes it a completely different word, for an automated method. Having the "wrong" word affects all n-grams containing that word for the BLEU scoring.

In order to achieve more quality in the translation assessment, we normalized all diacritics both in gold standard and in the submissions for Romanian. Results for the Medline en/ro test set are shown in Table 8, based on all sentences (en/ro) and only based on correctly aligned sentences (en/ro-OK). To this end, we wrote and used a simple sed-based script which brings the Romanian diacritics to the latest standard[7].

### 4.3 Manual evaluation

We performed manual evaluation of the primary runs (as identified by the participants) for each team and each language pair. The primary runs are compared to the reference translation and to each other, if more than one submission (from distinct teams) is available for the language pair. We computed pairwise combinations of translations either between two automated systems, or one automated system and the reference translation. The human validators were native speakers of the languages and were either members of the participating teams or colleagues from the research community. These are primary runs from each team:

- FOKUS: Medline en/ro run2;
- Hunter: Medline en/fr run2, EDP en/fr run2;
- LMU: Medline de/en run1, Medline en/de run1;
- TGF TALP UPC: Medline es/en run1, Medline fr/en run1, Medline pt/en run1;
- UFRGS: Medline en/es run1, Medline en/pt run1, Medline es/en run1, Medline pt/en run1;
- UHH-DS: Medline en/es run3, Medline en/pt run3, Medline en/ro run3, Medline es/en run3, Medline pt/en run3.

The validation task was carried out using the 3-way ranking task in our installation of the Appraise tool (Federmann, 2010).[8] For each pairwise

[6] For reference, the evolving standards for Romanian are discussed here http://kitblog.com/2008/10/romanian_diacritic_marks.html.

[7] The script is freely available at http://www.brainsignals.de/fixrodia.sh

[8] https://github.com/cfedermann/Appraise

comparison, we checked a total of 100 randomly-chosen sentence pairs. The validation consisted of reading the two sentences (A and B), i.e., translations from two systems or from the reference, and choosing one of the options below:

- A<B: when the quality of translation B was higher than A.
- A=B: when both translations had similar quality.
- A>B: when the quality of translation A was higher than B.
- Flag error: when the translation did not seem to be derived from the same input sentence. This is usually related to errors in corpus alignment.

We present the results for the manual evaluation of the Medline test sets in Table 9. Based on the number of times that a translation was validated as being better than another, we ranked the systems for each language as listed below:

- de/en: LMU = reference
- en/de: reference < LMU
- en/es: reference, UHH-DS < UFRGS
- en/fr: HunterNMT < reference
- en/pt: UHH-DS < UFRGS < reference;
- en/ro: UHH-DS < FOKUS < reference
- es/en: UHH-DS < UFRGS < TGF TALP UPC < reference
- fr/en: TGF TALP UPC < reference
- pt/en: UHH-DS < UFRGS < TGF TALP UPC < reference

Even though the LMU runs obtained one of the lowest BLEU scores (all of them less than 20 points), the primary run did score equally well or even better than the reference translation in the manual evaluation. The reason are misaligned sentences in the German reference. Automatic German translations on the other hand are most often correct in translation and alignment of content. Indeed, the quality of the German dataset was one of the lowest (cf. Section 2.1).

We present the results for the manual evaluation of the EDP test sets in Table 10. Based on the number of times that the submission was validated as being better than the reference translation, we ranked the two translations as follow:

- en/fr: Hunter < reference.

## 5 Discussion

In this section we present insight from the automatic and manual validations as well as on the quality of the translations.

### 5.1 Differences between manual and automatic evaluations

Similar to previous years, we did not notice any difference while ranking the teams for most language pairs regarding the automatic and manual evaluation of the translations. This year, the only significant difference we noticed was for the English translations, more specifically for es/en and pt/en pairs.

For es/en, the ranking order changed between the teams UFRGS and TGF TALP UPC. While the runs from the UFRGS teams achieved a higher BLEU score (43.31 vs. 40.49), our evaluators found the translations from the TGF TALP UPC team to be considerable better (79 vs. 7).

As for pt/en, the ranking of the teams changed from TGF TAP UPC < UHH-DS < UFRGS (automatic evaluation: 42.55 < 44.27 < 46.01) to UHH-DS < UFRGS < TGF TAP UPC (manual evaluation: 55 vs. 21 to UFRGS, 58 vs. 24 to UHH-DS). While no difference in ranking was observed between teams UHH-DS and UFRGS, in comparison to the automatic evaluation, team TGF TAP UPC moved from being the last ranked in the automatic evaluation to the best ranked one on the manual evaluation.

We can only hypothesize that the better BLEU scores that the UFRGS team obtained were probably due to better translation of particular concepts or due to using the same terms as in the reference translations. However, the TGF TAP UPC team could obtain higher quality of the manual translations using their Transformer architecture. The better performance of the TGF TAP UPC team could also have been due to the test set being included in the their training corpus, i.e. overlaps between Medline and the Scielo databases. While both teams trained on the Scielo corpus,

| **Languages** | **Runs (A vs. B)** | **Total** | **A>B** | **A=B** | **A<B** |
|---|---|---|---|---|---|
| de/en | LMU vs. reference | 75 | 29 | 14 | 32 |
| en/de | LMU vs. reference | 76 | 29 | 32 | 15 |
| en/es | UFRGS vs. reference | 86 | 37 | 23 | 26 |
|  | UFRGS vs. UHH-DS | 88 | 29 | 37 | 22 |
|  | reference vs. UHH-DS | 92 | 30 | 33 | 29 |
| en/fr | Hunter vs. reference | 92 | 14 | 13 | 65 |
| en/pt | UFRGS vs. reference | 86 | 6 | 43 | 42 |
|  | UFRGS vs. UHH-DS | 100 | 32 | 53 | 15 |
|  | reference vs. UHH-DS | 81 | 46 | 28 | 7 |
| en/ro | FOKUS vs. reference | 88/81 | 11/14 | 19/14 | 58/53 |
|  | FOKUS vs. UHH-DS | 100/97 | 57/55 | 31/27 | 12/15 |
|  | reference vs. UHH-DS | 88/85 | 80/78 | 6/6 | 2/1 |
| es/en | TGF TALP UPC vs. reference | 72 | 26 | 12 | 34 |
|  | TGF TALP UPC vs. UFRGS | 100 | 51 | 38 | 11 |
|  | TGF TALP UPC vs. UHH-DS | 98 | 79 | 12 | 7 |
|  | reference vs. UFRGS | 77 | 50 | 15 | 12 |
|  | reference vs. UHH-DS | 77 | 54 | 10 | 13 |
|  | UFRGS vs. UHH-DS | 100 | 45 | 24 | 31 |
| fr/en | TGF TALP UPC vs. reference | 85 | 24 | 19 | 42 |
| pt/en | TGF TALP UPC vs. reference | 89 | 25 | 26 | 38 |
|  | TGF TALP UPC vs. UFRGS | 100 | 55 | 24 | 21 |
|  | TGF TALP UPC vs. UHH-DS | 100 | 58 | 24 | 18 |
|  | reference vs. UFRGS | 87 | 42 | 22 | 23 |
|  | reference vs. UHH-DS | 87 | 52 | 28 | 7 |
|  | UFRGS vs. UHH-DS | 100 | 48 | 27 | 25 |

Table 9: Results for the manual validation for the Medline test sets. Values are absolute numbers (not percentages). They might not sum up to 100 due to the skipped sentences. Two evaluators (both participants) carried out the validation of the Romanian dataset and results from both of them are shown (separated by a slash).

| **Languages** | **Runs (A vs. B)** | **Total** | **A>B** | **A=B** | **A<B** |
|---|---|---|---|---|---|
| en/fr | Hunter vs. reference | 91 | 11 | 26 | 54 |

Table 10: Results for the manual validation for the EDP test sets. Values are absolute numbers (not percentages). They might not sum up to 100 due to the skipped sentences.

the UFRGS team reported that they tried to remove potential overlaps between Medline and Scielo (Soares and Becker, 2018). Overlaps of Medline and Scielo do not explain the lower BLEU scores obtained by the TGF TAP UPC team.

### 5.2 Differences across languages

Similar to previous years, comparison of results across languages did not provide any unexpected insight. The languages pairs which obtained higher BLEU scores (above 30 points), i.e. en/es, en/pt, es/en and pt/en, were also the ones for which more training data specific for the biomedical domain is available. Indeed, teams that participated with the same system for different language pairs obtained lower scores for those languages with fewer resources. This is the case of the scores of the TGF TAP UPC team for fr/en (up to 27 points) as opposed to the ones obtained for es/en and pt/en (more than 40 points).

We hope that recently released corpora, e.g. the BioRo corpus for Romanian (Mitrofan and Tufis, 2018), can boost performance of MT systems for these languages. However, more parallel corpora are certainly necessary not only for those languages that scored worst in this challenge, but also for the many other languages that we did not evaluate here. Unfortunately, open-access databases such as Scielo are not available for most languages. Nevertheless, the number of parallel abstracts in Medline are increasing and corpora derived from these are starting to being published, e.g. MeSpEn (Villegas et al., 2018).

### 5.3 Evolution of the performance in the last years

Compared to previous years, we found an improved BLEU score and improved manual evaluation for languages already considered in previous years, i.e. Portuguese, Spanish and French. This year we have considered Medline abstracts instead of Scielo ones.

However, the results are difficult to be directly compared to previous years given that test sets were from different sources for many of the language pairs. For the EDP test set, which can be considered very similar to last year's one, the Hunter team scored much better than their last participation both in terms of BLEU scores (17.50 vs. 23.24 for en/fr) and in the manual validation (0 to 93 vs. 11 to 54 in the manual validation).

The Medline test sets for es/en, pt/en, fr/en, en/es, en/pt and en/fr can be considered rather similar to the Scielo ones released for these language pairs in the two previous challenges (Bojar et al., 2016; Jimeno Yepes et al., 2017). From values below the 20 points in 2016, results from en/pt jumped to almost 40 in 2017 and over the latter (up to 43.14) this year. A similar increase was observed for en/es that increased more slowly from up to 33 points in 2016, up to 36 points in 2017 and up to 44 this year. On the other hand, not much improvement can be noticed from en/fr in 2016 (up to 22.75) to this year's best score (23.24). These values are also similar to the scores reported on another MEDLINE dataset in 2013 (Jimeno-Yepes and Névéol, 2013).

Regarding translations into English, for es/en, BLEU scores experienced an improvement from 37 to 43 points in the last year. However, the same could not be noticed for pt/en that remained rather constant around 41-43 points.

Finally, during our manual validation, we observed for the first time that the quality of some automatic translations was either equal or better than the reference translation. Two teams scored as good as the reference translation, namely, LMU for de/en and UHH-DS for en/es. Moreover, two teams scored higher than the reference translations, namely, LMU for en/de and UFRGS for en/es.

### 5.4 Quality of the automatic translations

Here we provide an overview of the quality of the translations and the common errors that we identified during the manual validation.

**English:** The English translations appeared in general to have improved qualitatively over prior year submissions. While in prior years translations often contained remnants of untranslated terms from the source language mixed into the translation, this problem was noted less often in this year's evaluations. In addition, systems appeared to make more effective use of capitalisation, avoiding translation of acronyms or attempting to translate an acronym semantically via its expansion.

In light of this overall improvement, a better translation is often decided by subtle, more precise choices of English words this year. For instance, an "increasing trend" is more precise as well as the more customary usage than "accentuating trend"; the "dissemination" of knowledge is likewise a better word choice than "diffusion" of knowledge. Similarly, a study objective "to assess" the level of something would be preferred to "to know" the level.

The automated systems maintained higher fidelity with the original texts than reference translations, with the latter often leaving out portions of the original sentence or restructuring information between contiguous sentences. Since the automated systems strive to translate the complete content of a sentence, they were in many cases perceived to be more accurate due to completeness, even where minor usage errors occurred.

An error that was observed regularly for Spanish to English translations in particular was the lack of a subject pronoun or insertion of a gendered pronoun ("He"/"She") at the start of a sentence where a demonstrative pronoun ("It", "This") would be more appropriate. As a pro-drop language, the source Spanish texts often lacked an overt subject; this subject needs to be introduced for the English translation to be fully grammatical but some systems appeared to struggle with this requirement.

Another error observed across different languages was the partial translation of multi-word biomedical terms. As an example, "upper digestive endoscopy" was translated as "high digestive endoscopy," where presumably "digestive endoscopy" was referenced from a biomedical dictionary but the word "high" was decoupled from the multi-word term. Although this error was less prevalent, its occurrences critically reduced the quality and crippled the scientific meaning of the translation.

**French:** The quality of translations for French seemed quite equivalent to last year, and varied from poor to good. A number of automatically translated sentences carried out the meaning of the original sentence properly, but were assessed as inferior to the reference for stylistic reasons, because they provided a more literal translation that mimicked the structure of the original sentence. Arguably, those sentences could be considered as useful to grasp the meaning of the original sentence. However, translation omissions were noted in long or complex sentences. For example, the phrase *"potential drug-drug and food-drug interactions"* was translated by *"interactions potentielles entre médicaments et médicaments"*, which does not account for food-drug interactions. A couple of recurring errors also observed in previous years are the lack of translation for acronyms and the erroneous choice of pronoun in translation. For example, *"they"* was systematically translated as *"ils"*, even when *"elles"* was the correct translation based on context.

The use of manual segmentation on the EDP corpus resulted in a number of single word "sentences" corresponding to the titles of the sections in structured abstracts, such as "Introduction:" or "Conclusion:". Unsurprisingly, the systems performed well on these isolated segments (except for one occurrence of *"Materials and Methods"* translated by *"Matériaux et Procédés"* instead of the usual *"Matériel et Méthodes"*), which may contribute to explain the number of instances where the automatic translation and manual reference were considered equivalent. It can be noted that dealing with section titles as isolated segments successfully ensured there were no translation errors linked to failure to identify the section words as isolated titles.

**German:** Interestingly, for 80% of the sentences automatic translation was evaluated equally good or even better than the German reference. This observed result has different reasons. Often the German reference translation is correct but either contains additional information or misses information present in the English source sentence while the automatic translation does not have this error. As previously mentioned this is strongly related to the frequent alignment errors present in the German dataset. In some cases validation was very difficult as both translations were very good but we still tried to differentiate between them. For instance, "thromboembolic complications" was translated to "thromboembolische Ereignisse" (events) in the reference and to "thromboembolische Komplikationen" (complications) in the automatic translation. In this case the evaluator scored LMU>Reference while also LMU=Reference would be possible.

In general, we only observe minor mistakes in automatic translation. Rarely, we find wrongly translated technical terms such as *cerebrum* wrongly translated to *Gebärmutter* (uterus). Often mistakes originate from a slight misuse of terms with the same overall meaning but different application in the medical domain. For instance, *soft tissue repair* was translated to *Weichteilsanierung* instead of *Weichteilrekonstruktion*, while the latter is the correct medical term. Similarly, *efficiency of medication* should be *Wirksamkeit von Medikamenten* instead of *Effizienz von Medikamenten*. Compared to submissions in 2017, we did not observe problems in syntax or grammar which could have caused misunderstanding the meaning of the sentence. This year, only the LMU team submitted reaults and already in 2017 their system did not have syntax problems.

**Portuguese:** We observed both minor and major mistakes in the automatic translations to Portuguese. We classified as minor those errors that did not compromise the overall understanding of the sentence and that were limited to orthography or minor grammatical errors. For instance, we found many wrong spaces separating compound words (e.g., *"amarelo- palha"*) and before commas or the final period (e.g., *"desafio médico , éticas"*). Further, translations from one particular system seemed to consistently start sentences with a lower case (e.g., *"manifestações clínicas de neurofibromatose tipo 1 são variáveis"*). Finally, other frequent minor mistakes were missing definite articles, such as in *"existem três variantes do osteocondroma extraesquelético : condromatose sinovial , condroma para-articular"* instead of *"a condromatose sinovial, o condroma para-articular"*.

We classified as major those mistakes that considerably compromised the understanding of the sentence. These were cases of discordance with number and gender for the adjectives, e.g., *"é um desafio médico , éticas e psicossociais"* instead of *"é um desafio médico, ético e psicossocial"*. There were also verbal discordances, such as in *"houve um caso que foram tratados"* instead of *"houve um caso que foi tratado"*. Further, we found many words that were not translated into Portuguese and remained in English, such as in the passages *"sem tratamento tumor-directed"*, *"Forty-seven casos"*. Also some acronyms were not correctly into Portuguese (e.g., *PET/CT* instead of *PET/TC*), but translations from one of the teams seem to have gotten most acronyms, biomedical terms and numbers right. Finally, given the differences in word ordering between English and Portuguese, this error occurred in passages such as *"pacientes queixa* instead of *"queixa dos pacientes"*.

Some translation were exactly like the reference translations, which makes us suspect that those abstracts could be included in the Scielo corpus used for training data by the systems. However, there were just a couple of such cases and these should not compromise overall evaluation. In spite of the above, we also found very good translations even for complex and long sentences and for biomedical terms with multiple modifiers, such as in *"esclerose múltipla secundária progressiva"*.

**Spanish:** Considering previous years of the challenge, the translations seem to improve and there are fewer issues compared to previous years. On the other hand, the issues we identified are similar to the ones identified in the Portuguese sets. As with the translations from Spanish into English, there were some cases of source words not being translated into Spanish, as in *"el estado fsico motor 20 Meter Shuttle Run Test"* and *"Substance-induced fueron"*.

Both types of methods seem to suffer from gender and number agreement for determiners as in *"La pulsos mejor en las"* and sometimes for verbs in terms of number as in *"Legendre describen el primer modelo"*, which might be misleading. We also found that some systems displayed a preference for starting sentences with lower case letters; however, different from the case in Portuguese, for the manually evaluated cases there no issues with acronyms or spaces between hyphenated words.

**Romanian:** This year there fewer participants than in the previous year. Especially regrettable is the absence of the top performers from the University of Edinburgh. The only team which participated last year as well is that of the University of Hamburg, which improved this year by using a training dataset subsampling heuristic in their SMT translator, but trailed again the NMT system in this task.

When manually comparing the translations, we have prefered the ones having the better grammar – for example "Diagnosticul precoce și tratamentul infecției **sunt asociate** (...)" was prefered to "Diagnosticul precoce şi tratamentul infecţiei **este asociat**(...)" for correct subject to verb agreement. Disturbing in translations is the occurence of words that have no correspondent in the original, for example "upon patients' arrival in the post anaesthesia care unit" translated as "*asupra* sosirii pacienților în unitatea de îngrijire a **tuberculozei**".

Totally incorrect translations were observed as well, such as “In this observational study, clinical data, vital signs and comfort parameters were collected from surgical patients who arrived in the PACU.” being translated into “În acest studiu observațional, date **contractuale**, semne vitale și parametri **portuari** au fost colectate de la pacienți **morali** care au sosit în PACU.”. In such a short sentence there are already three incorrectly translated words. A dictionary-based method would have done better, as there is also no ambiguity involved.

An interesting aspect was observed where the typical preprocessing leads to ambiguities. “MAP” (mean arterial pressure) changes to “map” (like in geographical map) and then is translated as such: “MAP and HR” was translated as “**Harta** şi *hr*”.

Another interesting and potentially dangerous error is the mistranslation of the time units. In one case “Haemofiltration was continued postoperatively in the ICU for another 48 h” was translated as “Haemofiltrarea a continuat postoperativ în ICU pentru încă 48 de **ani**” thus replacing hours with years.

In some cases the translations were marked equal because they were equally bad. In general, the intelligibility and fidelity of the translation was preferred to the form (grammar, smoothness, naturalness), and only for equal content the better form prevailed.

## 6 Conclusions

This was the third year we have organized the WMT biomedical shared task and we found that the performance has been increasing constantly. Improvements in results seem to be due to a variety of reasons, including more in-domain training data and the use of additional methods that consider transfer approaches and ensemble combination of methods.

From an evaluation perspective, we find that the results improve when we consider only sentences that were perfectly aligned instead of considering all the automatically aligned sentences. This shows some limitations on the quality of the automatically generated test sets. On the other hand, the comparative performance of the different participating systems remains the same.

For some of the languages considered, there were limitations in the quantity of available parallel abstracts. Recent publications include parallel corpora from the database that were previously used for obtaining our test sets. These new corpora include Medline parallel abstracts (Villegas et al., 2018) and full texts from Scielo (Soares et al., 2018). Therefore, manual translation for building the future test could be considered in the following editions of the challenge.

Finally, future improvements should also address problems reported by the participants regarding the current format of the test sets. In the three years of the challenge, we have used BioC as the format for data exchange, which seemed to cause some difficulties for sentence alignment. We will evaluate available formats for data exchange with the participants or inspired in other shared task in WMT.

## Acknowledgments

We would like to thank the participation of all teams in the challenge and the support of selected participants in the manual validation of the translations.

## References

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation (WMT16) at the Conference of the Association of Computational Linguistics*, pages 131–198.

Mirela-Stefania Duma and Wolfgang Menzel. 2018. Translation of biomedical documents with focus on Spanish-English. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Christian Federmann. 2010. Appraise: An open-source toolkit for manual phrase-based evaluation of translations. In *In LREC*.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88. Association for Computational Linguistics.

Cristian Grozea. 2018. Ensemble of translators with automatic selection of the best translation - the submission of FOKUS to the wmt 18 biomedical translation task -. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Matthias Huck, Dario Stojanovski, Viktor Hangya, and Alexander Fraser. 2018. LMU Munichs neural machine translation systems at WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Antonio Jimeno-Yepes and Aurélie Névéol. 2013. Effect of additional in-domain parallel corpora in biomedical statistical machine translation. In *Proceedings of the Fourth International Workshop on Health Text Mining and Information Analysis*.

Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Karin Verspoor, Ondrej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, Pavel Pecina, Roland Roller, Rudolf Rosa, Amy Siu, Philippe Thomas, and Saskia Trescher. 2017. Findings of the WMT 2017 Biomedical Translation Shared Task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247, Copenhagen, Denmark. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Lukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017. One model to learn them all. *CoRR*, abs/1706.05137.

Abdul Rafae Khan, Subhadarshi Panda, Jia Xu, and Lampros Flokas. 2018. Hunter NMT system for WMT18 biomedical translation task: Transfer learning in neural machine translation. In *Proceedings of the Third Conference on Machine Translation*, pages 1–2, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vladimir S. Lazarev and Serhii A. Nazarovets. 2018. Don't dismiss citations to journals not published in english. *Nature Correspondence*, 556:174.

Xiaoyi Ma. 2006. Champollion: A robust parallel text sentence aligner. In *In Proceedings of LREC-2006*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Maria Mitrofan and Dan Tufis. 2018. BioRo: The Biomedical Corpus for the Romanian Language. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Aurélie Névéol, Antonio Jimeno Yepes, Mariana Neves, and Karin Verspoor. 2018. Parallel corpora for the biomedical domain. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Mariana Neves, Antonio Jimeno Yepes, and Aurélie Névéol. 2016. The Scielo Corpus: a Parallel Corpus of Scientific Publications for Biomedicine. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Daniel Prieto. 2018. Make research-paper databases multilingual. *Nature Correspondence*, 560:29.

Felipe Soares and Karin Becker. 2018. UFRGS participation on the wmt biomedical translation shared task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Felipe Soares, Viviane Moreira, and Karin Becker. 2018. A Large Parallel Corpus of Full-Text Scientific Articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Juan Tao, Chengzhi Ding, and Yuh-Shan Ho. 2018. Publish translations of the best chinese papers. *Nature Correspondence*, 557:492.

Brian Tubay and Marta R. Costa-Jussà. 2018. Neural machine translation with the Transformer and multi-source romance languages for the biomedical wmt

2018 task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Marta Villegas, Ander Intxaurrondo, Aitor Gonzalez-Agirre, Montserrat Marimn, and Martin Krallinger. 2018. The MeSpEN resource for English-Spanish medical machine translation and terminologies: Census of parallel corpora, glossaries and term translations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

# An Empirical Study of Machine Translation for the Shared Task of WMT18

**Chao Bei, Hao Zong, Yiming Wang, Baoyong Fan, Shiqi Li, Conghu Yuan**
Global Tone Communication Technology Co., Ltd.
{beichao,zonghao, wangyiming, fanbaoyong, lishqi, yuanconghu}@gtcom.com.cn

## Abstract

This paper describes the Global Tone Communication Co., Ltd.'s submission of the WMT18 shared news translation task. We participated in the English-to-Chinese direction and get the best BLEU (43.8) scores among all the participants. The submitted system focus on data clearing and techniques to build a competitive model for this task. Unlike other participants, the submitted system are mainly relied on the data filtering to obtain the best BLEU score. We do data filtering not only for provided sentences but also for the back translated sentences. The techniques we apply for data filtering include filtering by rules, language models and translation models. We also conduct several experiments to validate the effectiveness of training techniques. According to our experiments, the Annealing Adam optimizing function and ensemble decoding are the most effective techniques for the model training.

## 1 Introduction

We participated in the WMT shared news translation task and focus on the English-to-Chinese direction. Our neural machine translation system is developed as transformer (Vaswani et al., 2017a) architecture and the toolkit we used is Marian (Junczys-Dowmunt et al., 2018). Since BLEU (Papineni et al., 2002) is the main ranking index for all submitted systems, we apply BLEU as the evaluation matrix for our translation system. We aim to verify whether the techniques we applied in the Encoder Decoder architecture of recurrent neural network(RNN) and attention mechanism (Bahdanau et al., 2014) are also positive for transformer architecture (Vaswani et al., 2017b) and the effectiveness of the data filtering.

For data preprocessing, the basic methods include Chinese word segmentation, tokenization, byte pair encoding(BPE) (Sennrich et al., 2015b). Besides, human rules and translation model are also involved for cleaning parallel data, as well as using language model for cleaning monolingual data. As to the techniques on model training, Annealing Adam (Denkowski and Neubig, 2017), back-translation (Sennrich et al., 2015a) and right-to-left reranking (Sennrich et al., 2016) which have proven to be effective in the Encoder Decoder model with RNN layer and attention mechanism are applied to verify whether these techniques in transformer architecture are also effective.

When comparing our baseline model, we show the increase in 5.57 BLEU scores of English to Chinese direction for news. And comparing the best score in last year, transformer architecture is more powerful than RNN with attention mechanism with 3.65 BLEU score improvement. However, not all the techniques we applied to RNN with attention mechanism are equally effective against transformer architecture, especially reranking by right-to-left model.

This paper is arranged as follows. We firstly describe the task and provided data information, then introduce the method of data filtering, including rules, language model and translation model. After that, we describe the techniques on transformer architecture and show the conducted experiments in detail, including data preprocessing, postprocessing and model architecture. At last, we analyse the results of experiments and draw the conclusion.

## 2 Task Description

The task focuses on bilingual text translation in news domain and the provided data is show in Table 1, including parallel data and monolingual data. The parallel data is mainly from News Commentary v13 (Tiedemann, 2012), UN Parallel Corpus V1.0 (Ziemski et al., 2016) and CWMT Cor-

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 340–344
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64031

| Direction | parallel data | monolingual data |
|---|---|---|
| en-zh | 22,587,593 | 9,061,023 |

Table 1: The number of provided data including parallel data and monolingual data.

pus, and monolingual we used is XMU corpus from CWMT Corpus. To compare with others in last year, WMT17 test set in English to Chinese direction is used as the development set to compare with the best score in last year.

## 3 Data Filtering

This section introduces the methods we used for data filtering in the news task. For this task, because we found that it is very difficult to make a significant improvement for training technique in a short time. Therefore, we pay more attention on the data filtering than exploring different training techniques. In this task we do the data filtering for both of the provided parallel sentences and the generated sentence from back translation.

### 3.1 Data Filtering through Rules

According to our observations the provided data has two types of noise: misalignment and translation error. One of the misalignment noise we found in the parallel corpus is that the translation only translates half or even a very small part of the source text. The translation error behaves like one punctuation repeated many times. Obviously language model cannot solve the problem of alignment or translation error from parallel sentences. It only evaluates the quality of the monolingual sentences. Thus, we clean up sentences with these problems with calculating the number of punctuation in both source sentence and target sentence. The parallel sentences where the difference between the number of punctuation of source and target sentences that exceeds the threshold A are removed. Besides, the sentences which contain punctuation more than threshold B will be removed because these sentences may appear as the table of contents or other sentences with some punctuation error. Here threshold A is named relative punctuation frequency threshold and threshold B is named absolute punctuation frequency threshold.

### 3.2 Data Filtering through Language Model

It has been proved that back translation (Sennrich et al., 2015a) is an effective way to improve the translation quality, especially in low-resource condition. In this task we firstly train an initial translation model(from Chinese to English) using transformer architecture, then we use this model to translate the provided monolingual Chinese data onto English and then get the generated synthetic data. To filter the generated synthetic data, we organize the filtering procedure as follows:

- Train two language models with Chinese and English monolingual data extracted from provided parallel corpus. To train the models we utilized the Marian toolkit, the model type of Marian is lm-transformer whose architecture is based on transformer.

- Calculate the cross entropy of each sentence with the trained language model in Chinese.

- Analyse the cross entropy, according to our observation, we removed the sentences with cross entropy higher than -30.971481 or lower than -299.529816. After this operation the number of remaining parallel sentences is 6,280,000 out of 9,061,023.

- Remove the duplicated sentences in the remaining 6,280,000. This operation further reduced the remaining sentences to 5,891,328.

- Remove the sentences that contain HTML tag such as "$\langle p\rangle\langle/p\rangle$","$\langle strong\rangle\langle/strong\rangle$", the remaining sentences then reduced to 4,981,288.

- Calculate the cross entropy of each translated English sentence with the trained English language model.

- Remove the sentences with cross entropy lower than -396.643829, the remaining parallel sentences further reduced from 4,981,288 to 4,975,094.

The reason why our filtering procedure is more complicated is that we believe the quality of the data can heavily affect the translation performance. We trained two language models to filter the synthetic data from both source text and target text. Through the above filtering procedure the synthetic data is reduced from 9,061,023 to 4,975,094.

### 3.3 Data Filtering Through Translation Model

Beside the generated synthetic data, we also suppose the provided parallel corpus is not clean enough to directly put into the training procedure. Since the language model cannot evaluate the quality of translation for parallel sentences which means that tow irrelevant or bad-translated sentences can't be distinguished through language model. Therefore, we use the rescorer tool of Marian to evaluate the parallel sentences in loss. In this case, we trained a translation model with the provided paralleled data, then we assume the translation model is generally correct and fix all the parameters in the model to calculate the cross entropy loss of each pair of provided parallel sentences. We remove the provided parallel sentences with cross entropy loss lower than -165.529449. This operation accompanies by the filter rules make the number of parallel sentences reduces from 22,587,593 to 17,969,826.

## 4 Optimizing transformer

The intuition for optimizing transformer is to try those optimizing methods which have proven to be effective in RNN architecture. According to our previous experiments right-to-left reranking, back translation synthetic data, Annealing Adam and ensemble decoding are the most effective approaches to improve the translation performance.

Right-to-left reranking means training a right-to-left model in target side. It can rerank the n-best translations and the expected averaged probabilities will be more robust for general evaluation. In this task, we reverse the target sentences and train the rights-to-left model.

Back translation is trying to improve the translation quality through data aspect. It is a simple but effective approach especially in low-resource condition. In this task, we have nearly 20 million parallel sentences from English to Chinese, but we are still trying to translate the Chinese monolingual data to construct the back translation data.

Annealing Adam is an optimizing function which is significantly faster than stochastic gradient descent with Annealing. Besides, it can also obtain a better performance in most cases. In this task we set the baseline with Annealing Adam optimizing function.

Ensemble decoding is trying to combine different models together to explore a better translation balance between different translation preference. The most common solution is to average the parameters of the latest server saved models during the training procedure. We can also combine models with different parameter initialization or even models with different hyper parameters. Normally to do ensemble decoding requires many different trained models. Therefore, it needs a lot of time and hardware resources which is the main reason that we only participate in one direction of the whole evaluation task. Unlike some other participants, we take a greedy ensemble strategy to combine our trained models instead of directly ensemble decoding them all. The greedy ensemble strategy firstly choose one model with the best single model BLEU score as the base model, and choose one model from the rest models again as the ensemble result to get a better BLEU score, then repeatedly choose one of the rest model to obtain a better BLEU until the BLEU doesn't increase.

## 5 Experiment

This section describes the all experiments we conducted and illustrate how we get the evaluation step by step.

### 5.1 Data pre-processing

In the news translation task we only focus on English to Chinese direction. Both of the parallel data and monolingual data are fully filtered at first. After that, we normalized the punctuation of English texts by normalize-punctuation.perl in Moses toolkit(Koehn et al., 2007) and normalized the punctuation of Chinese texts by converting the double byte character(DBC) to single byte character(SBC). We applied Jieba(Sun, 2012) as our Chinese word segmentation tool for segment Chinese text in both parallel data and monolingual data. For English text, tokenizer and truecase in Moses toolkit are applied. Finally, we applied BPE on both tokenized Chinese and English text.

### 5.2 Experiments setup

We describe all the experiment setups for this task in detail. The transformer baseline is trained with only parallel data, including CWMT corpus, UN Parallel Corpus V1.0 and News Commentary v13, after data preprocessing. We trained the baseline system not only in English to Chinese direction, but also in Chinese to English direction in order to translate the filtered monolingual data and do

| configuration | value |
|---|---|
| architecture | transformer |
| English vocabulary size | 40500 |
| Chinese vocabulary size | 50000 |
| word embedding | 512 |
| Encoder depth | 6 |
| Decoder depth | 6 |
| transformer heads | 8 |
| size of FFN | 2048 |

Table 2: The main model configuration. FFN means feed forward network.

| parameter | value |
|---|---|
| maximum sentence length | 100 |
| batch fit | true |
| learning rate | 0.0003 |
| label-smoothing | 0.1 |
| optimizer | Adam |
| learning rate warmup | 16000 |
| clip gradient | 5 |

Table 3: The training and decoding parameter.

| data | number |
|---|---|
| original data | 22,587,593 |
| cleaning by rules and TM | 17,969,826 |
| original synthetic data | 9,061,023 |
| synthetic sentences cleaning by LM | 4,981,288 |

Table 4: Cleaning parallel data and synthetic data. TM means translation model and LM means language model.

the parallel data filtering. During the training procedure the number of BPE merge operation is set to 40,000 for both English and Chinese. The hyperparameter of our baseline model configuration is shown in Table2 and the training parameter is in Table 3. After the baseline, we filter parallel data through rules and translation model. The relative punctuation frequency threshold and absolute punctuation frequency threshold we mentioned in section 3 is 5 and 15 respectively. We construct the synthetic data with back translation baseline model from Chinese to English. The synthetic data is firstly filtered by Chinese language model and then filtered by English language model. Table 4 shows the detail information about the data filtering.

In general, we trained 3 models to explore the effect of data filtering, which are: 1. baseline model with provided parallel sentences; 2. baseline model with parallel sentences filtered by rules and translation model; 3. baseline model with sentences mixed parallel sentences filtered by rules and translation model and synthetic sentences filtered by language model. Beside the baseline models, we trained four groups of translation model with fully filtered parallel data and synthetic data. Each model in the four groups is trained with different random seed and also apply Annealing Adam which get better performance compared with Adam. Therefore, we got 8 different translation models with the filtered data. We applied the greedy ensemble strategy to combine the 8 models and finally obtain the best translation performance on the development set with 3 models. Another, the right-to-left model in target side is also trained to rerank n-best translation of three best translation performance models.

## 6 Result and analysis

Table 5 shows the BLEU score we evaluated on development set. For data filtering, we observed that the methods improve the quality of sentences and get a better BLEU score. The methods can solve some problems of corpus quality. For model training techniques, back-translation is still the most effective method of improvement on 3.83-3.93 BLEU score. Annealing Adam has an improvement of BLEU score ranging from 0.04 to 0.36. The evaluation table shows that the higher BLEU score we get from the neural machine translation model, the smaller improvement can we get from Annealing Adam. When ensemble decoding, the greedy ensemble decoding strategy get the improvement on 0.56 BLEU score. However, when trying to decode our models ensemble with right-to-left reranking it did not improve the BLEU score as we expected.

Regard to the official evaluation we add one more post-processing step which is to convert all the SBC punctuation to DBC punctuation and it consequently further improved the BLEU score form 43.2 to 43.8.

## 7 Summary

We explored how to optimize the quality of machine translation in two different ways:1. through the data; 2 through the training and decoding approaches. In data aspect, we illustrated how we filter the provided parallel corpus through the trained

| **model** | **BLEU** |
|---|---|
| baseline with PS | 34.38 |
| + Annealing Adam | 34.74 |
| clean PS by rules and TM | 35.42 |
| + Annealing Adam | 35.56 |
| mix cleaned PS and SS cleaned by LM | 39.35 |
| + Annealing Adam | 39.39 |
| greedy ensemble decoding | 39.95 |
| r2l reranking | 39.91 |

Table 5: The BLEU score in character level for development set of English-to-Chinese direction. SS means synthetic sentences, TM means translation model, LM means language model and PS means parallel sentences. The greedy ensemble decoding means decoding the 8 models and finally obtain the best translation performance on development set with 3 models.

language model and trained translation model and showed the improvement of the data filtering, as well as constructing the synthetic through the back translation approach. In the training and decoding aspect, we applied transformer architecture as our main machine translation framework. To optimize it we utilized Annealing Adam optimize function and ensemble decoding. We also found that right to left reranking is not working according to our experiments.

## Acknowledgments

This work is supported by 2020 Cognitive Intelligence Research Institute[1] of Global Tone Communication Technology Co., Ltd.[2]

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Michael J. Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. *CoRR*, abs/1706.09733.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, Andr F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in c++. *arXiv preprint arXiv:1804.00344*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for WMT 16. *CoRR*, abs/1606.02891.

J Sun. 2012. jiebachinese word segmentation tool.

Jrg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. *CoRR*, abs/1706.03762.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *LREC*.

[1] http://www.2020nlp.com/

[2] http://www.gtcom.com.cn/

# Robust parfda Statistical Machine Translation Results

**Ergun Biçici**
ergun.bicici@boun.edu.tr
Department of Computer Engineering, Boğaziçi University
orcid.org/0000-0002-2293-2031
bicici.github.com

## Abstract

We build parallel feature decay algorithms (parfda) Moses statistical machine translation (SMT) models for language pairs in the translation task. parfda obtains results close to the top constrained phrase-based SMT with an average of 2.252 BLEU points difference on WMT 2017 datasets using significantly less computation for building SMT systems than that would be spent using all available corpora. We obtain BLEU upper bounds based on target coverage to identify which systems used additional data. We use PRO for tuning to decrease fluctuations in the results and postprocess translation outputs to decrease translation errors due to the casing of words. $F_1$ scores on the key phrases of the English to Turkish testsuite that we prepared reveal that parfda achieves $2nd$ best results. Truecasing translations before scoring obtained the best results overall.

## 1 Introduction

Statistical machine translation is widely prone to errors in text including encoding, tokenization, morphological variations and the mass they take, the size of the training and language model datasets used, and model errors. parfda is an instance selection tool based on feature decay algorithms (Biçici and Yuret, 2015) we use to select training and language model instances to build Moses phrase-based SMT systems to translate the test sets in the news translation task at WMT18 (WMT, 2018). As we work towards tools that can be used for multiple languages at the same time, we aim to obtain robust results for comparison and record the statistics of the data and the resources used. Our contributions are:

- a test suite for machine translation that is out of the domain of news task to take the chance of taking a closer look at the current status of SMT technology used by the task participants when translating 10 sentences taken from literary context in Turkish, which shows that parfda phrase-based SMT can obtain $2nd$ best results on this test set,
- parfda results for language pairs in the translation task and data statistics,
- comparison of processing alternatives for translation outputs to obtain better results,
- upperbounds on the translation performance using lowercased coverage to identify which models used data in addition to the parallel corpus,
- a set of rules that fix tokenization errors in Turkish using Moses' (Koehn et al., 2007) tokenization scripts.

We obtain parfda Moses phrase-based SMT (Koehn et al., 2007) results for the language pairs in both directions in the WMT18 news translation task, which include English-Czech (en-cs), English-Estonian (en-et), English-German (en-de), English-Finnish (en-fi), English-Russian (en-ru), and English-Turkish (en-tr). Building a language independent system that can perform well in translation tasks is a challenging task and SMT systems participating at WMT18 have been largely built dependent on the translation direction.

## 2 parfda

Parallel feature decay algorithms (parfda) (Biçici, 2016) parallelize feature decay algorithms (FDA), a class of instance selection algorithms that use feature decay, for fast deployment of accurate SMT systems. We use parfda to select parallel training data and language model (LM)

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 345–354
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64032

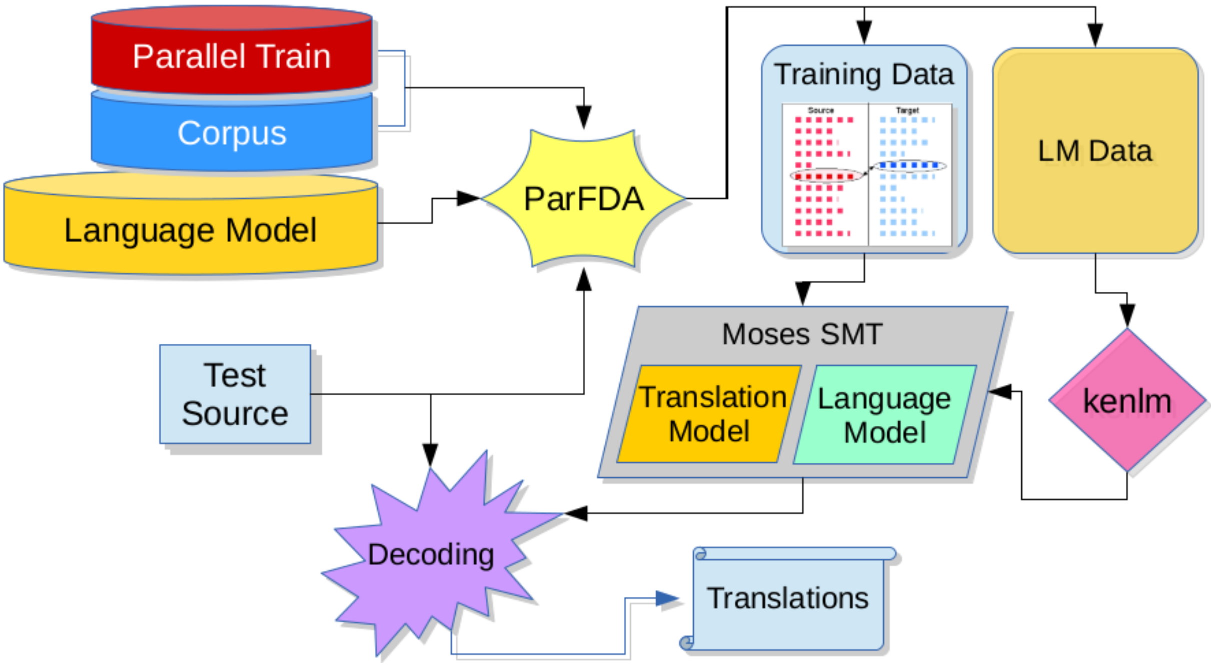


Figure 1: `parfda` Moses SMT workflow.

data for building SMT systems. `parfda` runs separate FDA5 (Biçici and Yuret, 2015) models on randomized subsets of the available data and combines the selections afterwards. Figure 1 depicts `parfda` Moses SMT workflow. The approach also obtained improvements using NMT (Poncelas et al., 2018).

We obtain transductive learning results since we use source sentences of the test set to select data. However, decaying only on the source test set features does not necessarily increase diversity on the target side thus we also decay on the target features that we already select. With the new `parfda` model, we select about 1.7 million instances for training data and about 15 million sentences for each LM data not including the selected training set, which is added later. Table 1 shows size differences with the constrained dataset (C). We use 3-grams to select training data and 2-grams for LM data. TCOV lists the target coverage in terms of the 2-grams of the test set. We also use CzEng17 (Bojar et al., 2016) for en-cs and SETIMES2 (Tiedemann, 2009) for en-tr.

We set the maximum sentence length to 126 and train 6-gram LM using `kenlm` (Heafield et al., 2013). For increasing the robustness of the optimization results, we use PRO (Section 2.1) and we use varying n-best list size. For word alignment, we use mgiza (Gao and Vogel, 2008) where GIZA++ (Och and Ney, 2003) parameters set max-fertility to 10, the number of iterations to 7,3,5,5,7 for IBM models 1,2,3,4, and the HMM model, and learn 50 word classes in three iterations with the mkcls tool during training. The development set contains up to 4000 sentences randomly sampled from previous years' development sets (2011-2017) and remaining come from the development set for WMT18. Table 2 lists the coverage of the test set.

### 2.1 Robust Optimization Results with PRO

Pairwise ranking optimization (PRO) (Hopkins and May, 2011) is found to obtain scores that monotonically increase, with results that are at least as good as MERT (Och, 2003), and with a standard deviation that is three times lower than MERT. We use PRO for tuning to obtain robust results due to fluctuating scores with MERT. PRO tuning performance graph is compared with MERT performance plot in Figure 2. We used monotonically increasing n-best list size at the start to increase robustness by using multiples of 50 until the 8th iteration, 350 every 10th, and 150 in the remaining. We only need 4 iterations to find parameters whose tuning score reach $1\%$ close to the best tuning parameter set score (Figure 3).

### 2.2 Testsuite for en-tr and tr-en

We prepared an SMT test suite that is out of the domain of news translation task to take a closer

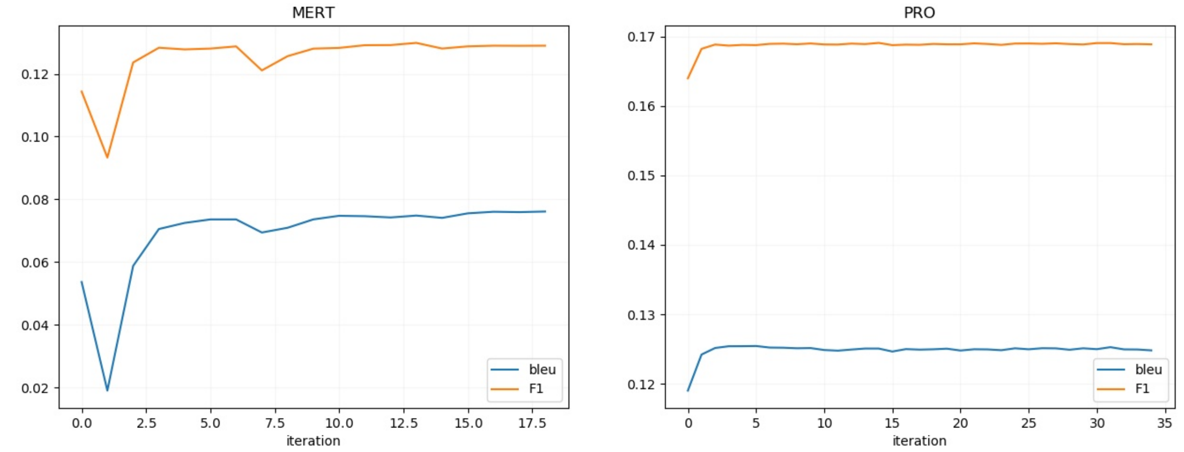


Figure 2: Comparison of MERT and PRO tuning on en-tr using results from 2017 and 2018 respectively.

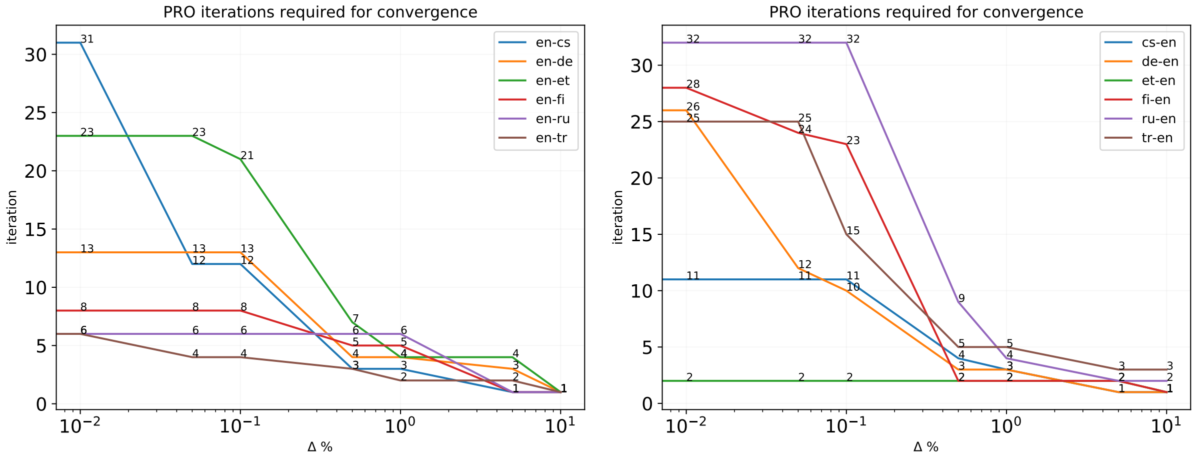


Figure 3: The number of iterations PRO would need to reach $\Delta\%$ close to the best tuning score.

look at the current status of SMT technology used by the task participants to translate 10 sentences taken from literary context in Turkish. The sentences and their translations are provided in Appendix A.

Table 3 details the testsuite results on en-tr and tr-en where the best translations of parfda are selected based on their BLEU (Papineni et al., 2002) and $F_1$ (Biçici, 2011) scores:

| | |
|---|---|
| en-tr | lctc 1 align |
| en-tr ts | lctc 1 align |
| tr-en | tc 2 align |
| tr-en ts | tc 1 align |

where tc and lctc are defined in Section 2.3.

We count tokens of translation as non-translation when they are found in the test source, are not a number or punctuation, and are considered by the SMT model's phrase table or the lexical translation table as a token whose translation differs from the source token. We have access to the lexical tables of parfda SMT models and among the tr-en lctc entries (Table 4), 2.7% contain a translation the same as the source. According to the testsuite results using translations from task participants, only RWTH and parfda contained non-translations and RWTH had only a token non-translated. The scores for up to $n$-grams in Table 12 show that alibaba.5744 achieves the best results in en-tr and online-B achieves the best results in tr-en in all scores. When we look at some of the OOV tokens in en-tr, we observe that lowercasing and then truecasing might help.

We identified 5 key phrases for both en-tr and tr-en that we would like to see translated correctly (Table 5). Some are trimmed to make them closer to their root form so that suffixes can be added without decreasing identification rates. Appendix A presents $F_1$ scores based on the identification of them in the translations. We see that even though parfda achieves the lowest scores in BLEU, on the key phrases, it provides the 2nd

| $S \rightarrow T$ | Data | Training Data |  |  |  | LM Data |  |
|---|---|---|---|---|---|---|---|
|  |  | #word S (M) | #word T (M) | #sent (K) | TCOV | #word (M) | TCOV |
| en-cs | C | 618.2 | 548.9 | 43274 | 0.749 | 1357.0 | 0.857 |
| en-cs | parfda | 83.6 | 73.9 | 1777 | 0.663 | 416.2 | 0.809 |
| cs-en | C | 548.9 | 618.2 | 43274 | 0.855 | 11100.4 | 0.948 |
| cs-en | parfda | 78.1 | 88.1 | 1773 | 0.8 | 502.2 | 0.896 |
| en-de | C | 580.2 | 548.0 | 24914 | 0.791 | 3078.9 | 0.892 |
| en-de | parfda | 96.6 | 91.4 | 1776 | 0.74 | 467.3 | 0.839 |
| de-en | C | 548.0 | 580.2 | 24914 | 0.859 | 11100.4 | 0.941 |
| de-en | parfda | 90.4 | 94.9 | 1776 | 0.82 | 509.1 | 0.893 |
| en-et | C | 20.4 | 27.0 | 1073 | 0.422 | 1197.1 | 0.772 |
| en-et | parfda | 27.0 | 20.4 | 1072 | 0.422 | 375.8 | 0.654 |
| et-en | C | 27.0 | 20.4 | 1073 | 0.673 | 11100.4 | 0.943 |
| et-en | parfda | 20.4 | 27.0 | 1072 | 0.673 | 416.4 | 0.883 |
| en-fi | C | 52.3 | 72.3 | 2846 | 0.45 | 1536.4 | 0.743 |
| en-fi | parfda | 53.4 | 38.6 | 1598 | 0.438 | 468.0 | 0.676 |
| fi-en | C | 72.3 | 52.3 | 2846 | 0.725 | 11100.4 | 0.943 |
| fi-en | parfda | 37.2 | 50.8 | 1550 | 0.715 | 473.2 | 0.888 |
| en-ru | C | 172.6 | 202.3 | 8766 | 0.739 | 9643.5 | 0.916 |
| en-ru | parfda | 69.3 | 53.0 | 1712 | 0.684 | 561.3 | 0.83 |
| ru-en | C | 202.3 | 172.6 | 8766 | 0.844 | 11100.4 | 0.95 |
| ru-en | parfda | 61.7 | 71.6 | 1764 | 0.816 | 489.6 | 0.903 |
| en-tr | C | 4.6 | 5.1 | 208 | 0.352 | 4026.5 | 0.824 |
| en-tr | parfda | 5.1 | 4.6 | 207 | 0.352 | 474.5 | 0.72 |
| tr-en | C | 5.1 | 4.6 | 208 | 0.569 | 11100.4 | 0.936 |
| tr-en | parfda | 4.6 | 5.1 | 207 | 0.569 | 442.1 | 0.877 |

Table 1: Statistics for the training and LM corpora in the constrained (C) setting compared with the parfda selected data. #words is in millions (M) and #sents in thousands (K). TCOV is target 2-gram coverage.

|  | SCOV |  |  |  |  | TCOV |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
|  | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| en-cs | 0.9635 | 0.8612 | 0.6212 | 0.3305 | 0.1334 | 0.9718 | 0.7587 | 0.4042 | 0.165 | 0.0553 |
| en-de | 0.9657 | 0.8646 | 0.627 | 0.3261 | 0.1253 | 0.9362 | 0.7939 | 0.5254 | 0.2534 | 0.0904 |
| en-et | 0.8912 | 0.6821 | 0.3717 | 0.1345 | 0.0376 | 0.8095 | 0.4308 | 0.1664 | 0.0528 | 0.0136 |
| en-fi | 0.9135 | 0.7339 | 0.4477 | 0.1889 | 0.0619 | 0.834 | 0.4595 | 0.1895 | 0.0612 | 0.0165 |
| en-ru | 0.9682 | 0.8683 | 0.6444 | 0.3581 | 0.1567 | 0.9703 | 0.78 | 0.4572 | 0.2076 | 0.0812 |
| en-tr | 0.8286 | 0.5817 | 0.2807 | 0.0905 | 0.0226 | 0.7944 | 0.3613 | 0.12 | 0.0282 | 0.006 |

Table 2: Test set SCOV and TCOV for $n$-grams.

best in en-tr among 9 models and 4th best among 6 in tr-en. Key phrase identification is important since when scores are averaged, important phrases that are missing only decrease the score by $\frac{1}{|p|N_{|p|}}$ for BLEU calculation for a phrase of length $|p|$ over $N_{|p|}$ phrases with length $|p|$.

## 2.3 Comparing Text Processing Settings for SMT

Experiment management system (EMS) (Koehn, 2010) of Moses prepares translations as follows:

truecase input
→ translate input
→ clean output (XML tags)
→ detruecase output

Truecasing updates the casing of words according to the most common form observed in the whole training corpus. EMS does not truecase the translations of an SMT model when training data are already truecased. However, each casing of words are a different entry in the phrase table and the casing we are interested in might be missing in the translations. Therefore, truecasing (tc) before detruecasing makes sense.

The casing of the text affects the number of tokens in the data sets. A casing of a token might appear in the phrase table but not its lowercased (lc) version. In EMS, truecasing is applied on the input. We experiment with truecasing lowercased text (lctc) to decrease the number of out-of-vocabulary words in the translations and to reduce the number of unique $n$-grams, dataset sizes, and the binary LM size by about 2%.

We process tokenized Turkish text using a set of rules since Moses' (Koehn et al., 2007) tokenization scripts can encounter tokenization errors in Turkish. A simpler approach was also tried for fixing tokenization of Turkish by removing space for unbalanced single quotes (Ding et al., 2016). Additionally, we retain the casing of the

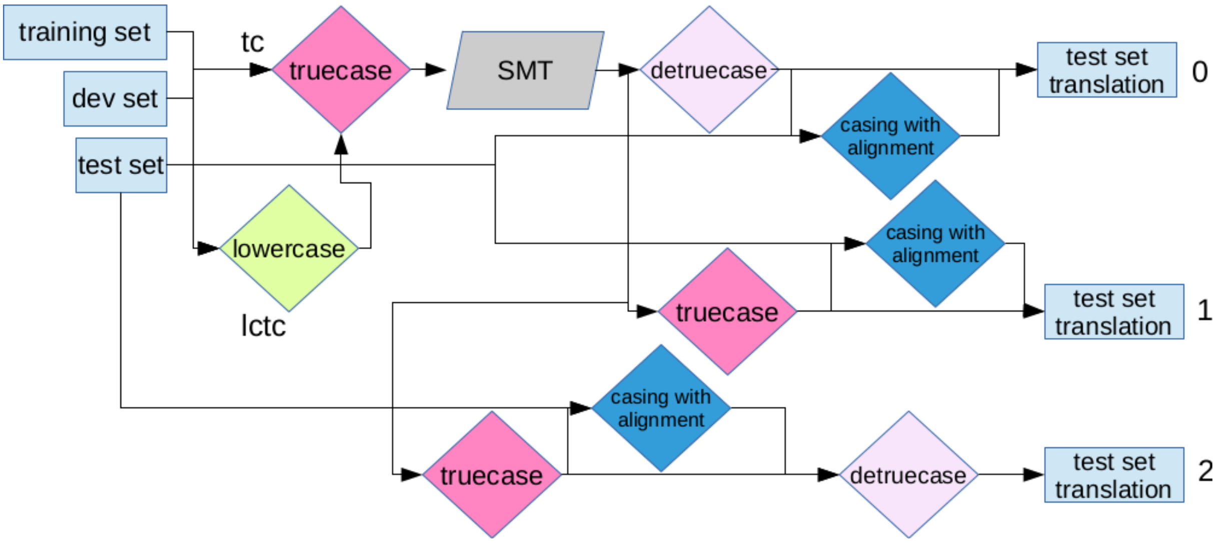


Figure 4: SMT text processing comparison of truecasing after lowercasing (lctc) and truecasing (tc).

| testset | setting | % sent. OK | non-trans. | non-trans. % | not in PT/lex | oov | % oov | BLEU | $F_1$ |
|---|---|---|---|---|---|---|---|---|---|
| en-tr | `tc` 1 align | 60.9 | 1941 | 18.81 | 22 | 1324 | 12.8 | 0.0746 | 0.1302 |
| | `lctc` 1 align | 61.1 | 1933 | 18.74 | 0 | 1290 | 12.5 | 0.0883 | 0.1406 |
| en-tr ts | `lctc` 1 align | 60.0 | 5 | 3.4 | 0 | 4 | 2.7 | 0.051 | 0.105 |
| tr-en | `tc` 2 align | 38.9 | 3549 | 20.7 | 23 | 2643 | 15.4 | 0.1192 | 0.1708 |
| | `lctc` 2 align | 44.8 | 3028 | 17.6 | 12 | 2132 | 12.4 | 0.1055 | 0.1567 |
| tr-en ts | `tc` 1 align | 20.0 | 32 | 17.7 | 0 | 24 | 13.3 | 0.0 | 0.1011 |

Table 3: Best performing en-tr and tr-en translation results detailed with their types of errors.

| | setting | N (lexical) | % ($w_S == w_T$) |
|---|---|---|---|
| en-tr | `tc` | 928K | 2.91 |
| | `lctc` | 890K | 2.68 |
| tr-en | `tc` | 891K | 3.02 |
| | `lctc` | 860K | 2.74 |

Table 4: Lexical translation table comparison.

| | phrase | count |
|---|---|---|
| en-tr | Türk Dil Kurumu | 6 |
| | Türkçe Sözlü | 4 |
| | Yazım Kılavuzu Çalışma Grubu | 3 |
| | Yazım Kılavuzu | 7 |
| | yazım kural | 4 |
| tr-en | Turkish Language Institution | 6 |
| | Turkish Language | 6 |
| | Turkish Dictionary | 4 |
| | Working Group | 3 |
| | Writing Manual | 4 |

Table 5: Key phrases we look for in the translations.

test source sentences using the word alignment information (Ding et al., 2016). Using alignment information is more complicated since not all alignments are 1-to-1. We also experiment with finding the casing of the input words in the development and test sets according to the form found in the translation tables to replace them before decoding. Figure 4 compares `tc` and `lctc` approaches to text processing for SMT. Both can use the alignment information for casing words.

Table 6 compares the results using translations that contain the alignment information and the unknown words where `tc` 0 is the baseline. The additional Moses decoder parameter is `--print-alignment-info`. We obtain the highest en-tr score using the alignments for casing but scores decrease for en-de and de-en. For which translation directions it helps can be seen in the `lctc` 0 row. The difference between the base and the lowercased results are the gain we can achieve if we fix casing accordingly. Using `tc` translation as a start, the gain on average is about 1.1 BLEU points (0.011 BLEU). The best setting overall is `tc` 2. The largest room for improvement with `lctc` `lc` BLEU results are for cs-en and tr-en.

| BLEU | | cs-en | de-en | et-en | fi-en | ru-en | tr-en | en-cs | en-de | en-et | en-fi | en-ru | en-tr | sum |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| tc 0 | base | **0.2296** | **0.3332** | **0.1766** | **0.1536** | 0.247 | **0.1286** | **0.1567** | **0.2669** | **0.1182** | **0.1016** | 0.1934 | 0.0823 | 2.1877 |
| | align | 0.2134 | 0.1851 | 0.1684 | 0.1427 | 0.233 | 0.1269 | 0.1455 | 0.1637 | 0.1138 | 0.095 | 0.1821 | 0.083 | 1.8526 |
| tc 1 | | 0.2239 | 0.318 | 0.1718 | 0.1493 | 0.2464 | 0.1269 | 0.1506 | 0.2543 | 0.1117 | 0.0961 | 0.1918 | 0.0824 | 2.1232 |
| | align | 0.2127 | 0.1845 | 0.1682 | 0.1416 | 0.2329 | 0.1261 | 0.1455 | 0.1625 | 0.1137 | 0.0948 | 0.182 | 0.083 | 1.8475 |
| tc 2 | | 0.2295 | 0.3331 | 0.1765 | 0.1535 | **0.2526** | 0.1284 | 0.1566 | 0.2669 | 0.1182 | 0.1013 | **0.1989** | 0.0823 | **2.1978** |
| | align | 0.2136 | 0.1945 | 0.1681 | 0.1458 | 0.2347 | 0.1273 | 0.1469 | 0.1645 | 0.1151 | 0.0965 | 0.1827 | 0.0816 | 1.8713 |
| | lc | 0.2394 | 0.3447 | 0.184 | 0.1614 | **0.2632** | 0.1373 | 0.1622 | 0.2727 | **0.1216** | 0.1053 | **0.2048** | 0.089 | 2.2856 |
| lctc 0 | | 0.1869 | 0.2736 | 0.1523 | 0.129 | 0.1901 | 0.1066 | 0.1339 | 0.1251 | 0.0975 | 0.0875 | 0.1671 | 0.0616 | 1.7112 |
| | align | 0.2113 | 0.2443 | 0.1661 | 0.1266 | 0.211 | 0.1043 | 0.1448 | 0.1603 | 0.1114 | 0.0941 | 0.1644 | 0.0882 | 1.8268 |
| lctc 1 | | 0.1817 | 0.2602 | 0.1484 | 0.1252 | 0.1901 | 0.1064 | 0.1294 | 0.1149 | 0.0934 | 0.0828 | 0.1656 | 0.0639 | 1.662 |
| | align | 0.2105 | 0.2437 | 0.1658 | 0.1247 | 0.211 | 0.1032 | 0.1447 | 0.1585 | 0.1113 | 0.094 | 0.1642 | **0.0887** | 1.8203 |
| lctc 2 | | 0.1896 | 0.275 | 0.1552 | 0.1303 | 0.1974 | 0.1083 | 0.1369 | 0.127 | 0.1005 | 0.0887 | 0.1712 | 0.0632 | 1.7433 |
| | align | 0.2121 | 0.2583 | 0.1663 | 0.1304 | 0.2125 | 0.1055 | 0.1468 | 0.1611 | 0.1125 | 0.0954 | 0.1648 | 0.0871 | 1.8528 |
| | lc | **0.2445** | **0.3452** | **0.1871** | **0.1634** | 0.2506 | **0.1402** | **0.1651** | **0.2781** | 0.1212 | **0.1056** | 0.1803 | **0.0978** | 2.2791 |

Table 6: parfda tokenized and cased results with different text processing settings. Baseline is tc 0 (in *italic*). **bold** lists the best for a translation direction.

| BLEU | cs-en | de-en | et-en | fi-en | ru-en | tr-en | en-cs | en-de | en-et | en-fi | en-ru | en-tr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| parfda 2018 | 0.2322 | 0.3343 | 0.1741 | 0.1547 | 0.2485 | 0.1267 | 0.1529 | 0.2674 | 0.1203 | 0.0968 | 0.1970 | 0.0821 |
| parfda 2018 $F_1$ | 0.2551 | 0.344 | 0.2123 | 0.1936 | 0.2685 | 0.1768 | 0.1921 | 0.2891 | 0.1613 | 0.1494 | 0.2214 | 0.1314 |
| TopC NMT 2018 lc | 0.348 | 0.499 | 0.315 | 0.258 | 0.358 | 0.291 | 0.266 | 0.489 | 0.258 | 0.192 | 0.348 | 0.207 |
| TopC NMT 2018 | 0.339 | 0.484 | 0.307 | 0.249 | 0.349 | 0.28 | 0.26 | 0.483 | 0.252 | 0.182 | 0.348 | 0.20 |
| - parfda | 0.1068 | 0.1497 | 0.1329 | 0.0943 | 0.1005 | 0.1533 | 0.1071 | 0.2156 | 0.1317 | 0.0852 | 0.1510 | 0.1179 |
| avg diff lc | 0.1288 | | | | | | | | | | | |

Table 7: parfda results compared with the top results in WMT18 and their difference.[1]

parfda results at WMT18 are in Table 7 using BLEU over tokenized text. We compare with the top constrained submissions at WMT18 in Table 7 and at WMT17 in Table 8. [2] Performance compared with the top constrained (TopC) phrase-based SMT improved to 2.252 in 2017 from 3 BLEU points difference on average compared with WMT16 results, which is likely due to the new parfda model and phrase-based SMT being less common in 2017. parfda Moses SMT system can obtain 0.6 BLEU points close to the top result in Finnish to English translation in 2017. All top models use NMT in 2018 and most use backtranslations, which means that their TCOV is upper bounded by LM TCOV.

## 3 Translation Upper Bounds with TCOV

We obtain upper bounds on the translation performance based on the target coverage (TCOV) of $n$-grams of the test set found in the selected parfda training data (Bicici, 2016) but using lowercased text this time. For a given sentence $T'$, the number of OOV tokens are identified:

$$OOV_r \quad = \quad \texttt{round}((1 - \text{TCOV}) * |T'|) \quad (1)$$

where $|T'|$ is the number of tokens in the sentence. We obtain each bound using 500 such instances and repeat for 10 times. TCOV BLEU bound is optimistic since it does not consider reorderings in the translation or differences in sentence length. Each plot in Table 9 locates TCOV BLEU bound obtained from each $n$-gram and from $n$-gram TCOVS combined up to and including $n$ and ■ locates the parfda result and ★ locates the top constrained result. In en-de and en-tr, the top model achieves a higher score than the TCOV BLEU bound, which indicates that data additional to the constrained training data was used. In both, backtranslations were used.

## 4 Conclusion

We use parfda for selecting instances for building SMT systems using less computation overall and results at WMT18 provides new data about using the current phrase-based SMT technology towards rapid SMT system development. Our data processing experiments show that lowercasing and then truecasing data can improve SMT models and translation results provided that we can find the casing correctly and truecasing translations before scoring can improve the results. Our

[2] Due to different tokenization rules used by mteval-v14.pl in matrix.statmt.org, parfda BLEU scores are higher than the scores in Table 6.

[2] We use the results from matrix.statmt.org.

| BLEU | cs-en | de-en | fi-en | lv-en | ru-en | tr-en | en-cs | en-de | en-fi | en-lv | en-ru | en-tr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| parfda 2017 | 0.2276 | 0.2613 | 0.1987 | 0.1549 | 0.2812 | 0.1172 | 0.1381 | 0.1851 | 0.1282 | 0.1303 | 0.218 | 0.097 |
| TopC NMT 2017 | 0.309 | 0.351 | | 0.19 | 0.308 | 0.179 | 0.228 | 0.283 | 0.207 | 0.183 | 0.298 | 0.165 |
| - parfda | 0.0814 | 0.0897 | | 0.0351 | 0.0268 | 0.0618 | 0.0899 | 0.0979 | 0.0788 | 0.0527 | 0.0800 | 0.0680 |
| avg diff | 0.0693 | | | | | | | | | | | |
| TopC phrase 2017 | 0.265 | | 0.205 | 0.168 | 0.315 | 0.126 | 0.191 | 0.216 | 0.145 | 0.142 | 0.253 | 0.098 |
| - parfda | 0.0374 | | 0.0063 | 0.0131 | 0.0338 | 0.0088 | 0.0529 | 0.0309 | 0.0168 | 0.0127 | 0.035 | 0.001 |
| avg diff | 0.02252 | | | | | | | | | | | |

Table 8: parfda results compared with the top results in WMT17 and their difference.

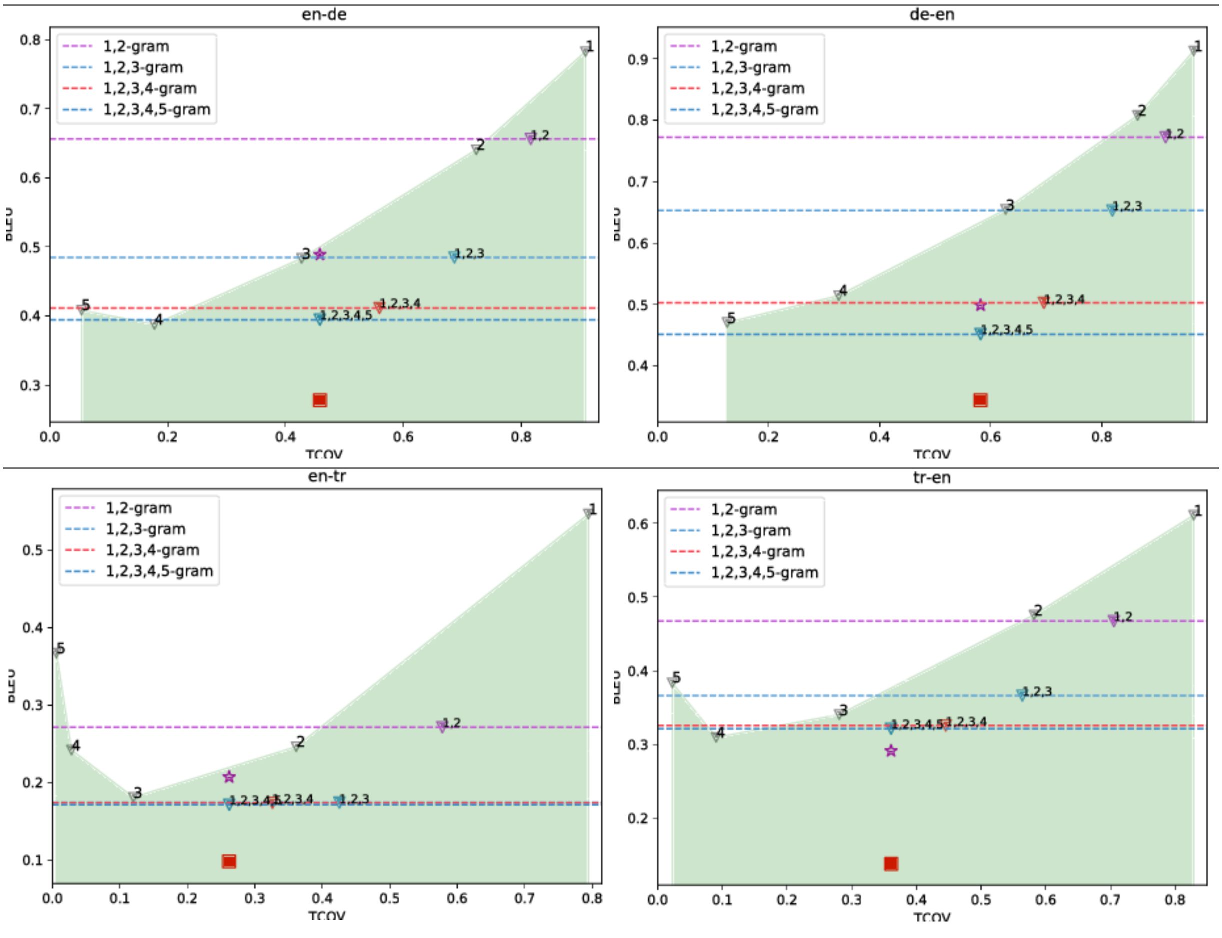


Table 9: parfda results (🟥) and $OOV_r$ TCOV BLEU upper bounds for de and tr.

method of tuning with PRO provides robust results and the BLEU bounds we obtain show which systems used additional training data. We are often interested to conserve the semantic content in the translations and parfda Moses phrase-based SMT achieves $2nd$ best results on the tr-en testsuite in our evaluations with key phrases.

## Acknowledgments

The research reported here received financial support in part from the Scientific and Technological Research Council of Turkey (TÜBİTAK) without contribution to the content nor responsibility thereof. We also thank the reviewers' comments.

## References

2018. *Proc. of the Third Conference on Machine Translation*. Association for Computational Linguistics, Brussels, Belgium.

Ergun Biçici. 2011. *The Regression Model of Machine Translation*. Ph.D. thesis, Koç University. Supervisor: Deniz Yuret.

Ergun Biçici and Deniz Yuret. 2015. Optimizing instance selection for statistical machine translation with feature decay algorithms. *IEEE/ACM Transactions On Audio, Speech, and Language Processing (TASLP)*, 23:339–350.

Ergun Bicici. 2016. ParFDA for instance selection for statistical machine translation. In *Proc. of the*

*First Conference on Statistical Machine Translation (WMT16)*, pages 252–258, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. *CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered*. Springer International Publishing, Cham.

Shuoyang Ding, Kevin Duh, Huda Khayrallah, Philipp Koehn, and Matt Post. 2016. The jhu machine translation systems for wmt 2016. In *Proceedings of the First Conference on Machine Translation*, pages 272–280, Berlin, Germany. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. 2008. *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, chapter Parallel Implementations of Word Alignment Tool. Association for Computational Linguistics.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362. Association for Computational Linguistics.

Philipp Koehn. 2010. An experimental management system. *The Prague Bulletin of Mathematical Linguistics*, 94:87–96.

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. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proc. of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. *Association for Computational Linguistics*, 1:160–167.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, USA.

Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. 2018. Feature decay algorithms for neural machine translation. In *Proc. of the 21th Conference of the European Association for Machine Translation (EAMT)*, Spain.

Jorg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

# A en-tr and tr-en Testsuite Sentences

| | English |
|---|---|
| 1 | The Turkish Language Institution’s Turkish Dictionary and Writing Manual Working Group has reached the twenty-seventh edition of the Written Manual. |
| 2 | To say upfront; to make the writing rules lasting and rooted to reach unity in writing, no changes have been made in the established rules in this edition as well. |
| 3 | Turkish Language Institution’s principles of taking the manual published in 1996 as the basis for traditional writing structures and the principle of taking the middle path to end discussions in writing are also followed in this edition. |
| 4 | This edition of the Writing Manual is prepared in parallel with the last edition of the Turkish Dictionary and new issues that emerge with the use of language are attached to rules and issues that were not addressed in previous editions are converted into rules. |
| 5 | Turkish Dictionary and Writing Manual Working Group reviews hundreds of examples about every rule and considers usage statistics with the principle of viewing writing as habit and custom. |
| 6 | After using Latin alphabet for more than eighty years we can say that Turkish writing has been traditionalized against some Latin sourced problems. |
| 7 | Without a doubt, Turkish Language Institution’s work for 80 years has played an important role in this. |
| 8 | The task of preparing, writing, and distributing writing manual is given to Turkish Language Institution according to the ç subitem of the 10th item in 664 numbered decree law based on the 134th item in the constitution. |
| 9 | Since its establishment, Turkish Language Institution has been trying to fulfill its duty on determining writing rules and publication of writing manuals. |
| 10 | Turkish Language Institution Turkish Dictionary and Writing Manual Working Group has diligently worked on each writing rule and writing to end discussions in writing and to spread writing rules and styles that everybody will accept and use. |

| | Turkish |
|---|---|
| 1 | Türk Dil Kurumu Güncel Türkçe Sözlük ve Yazım Kılavuzu Çalışma Grubunun yaptığı çalışmalarla Yazım Kılavuzu yirmi yedinci baskısına ulaşmış bulunuyor. |
| 2 | Hemen belirtelim; yazım kurallarını kalıcı hâle getirmek, kökleştirmek, böylece yazımda birliği sağlamak amacıyla bu baskıda da yerleşmiş kurallarla ilgili olarak herhangi bir değişikliğe gidilmemiştir. |
| 3 | Türk Dil Kurumunun 1996 yılında yayımladığı kılavuzda yazımı gelenekleşmiş biçimleri esas alma, yazımdaki tartışmalara son verme amacıyla yazımda orta yolun tutturulması ilkesi bu baskıda da gözetilmiştir. |
| 4 | Yazım Kılavuzu’nun bu baskısı Türkçe Sözlük’ün son baskısıyla eş güdüm içerisinde hazırlanmış, dilde yaşanan gelişmeler sonucunda ortaya çıkan yazımla ilgili yeni sorunlar bir kurala bağlanmış, önceki baskılarda değinilemeyen konular yazım kuralı hâline getirilmiştir. |
| 5 | Güncel Türkçe Sözlük ve Yazım Kılavuzu Çalışma Grubu, her kuralla ilgili yüzlerce örneği gözden geçirirken yazımın bir alışkanlık ve gelenek olduğu ilkesiyle kullanım sıklıklarını göz önünde bulundurmuştur. |
| 6 | Seksen yılı aşkın bir süredir kullanmakta olduğumuz Latin kaynaklı Türk yazısıyla kimi sorunlara karşın artık yazımın gelenekleştiğini söyleyebiliriz. |
| 7 | Bunda hiç kuşkusuz, Türk Dil Kurumunun seksen yıla ulaşan çalışmaları önemli bir rol oynamaktadır. |
| 8 | Ülkemizde yazım kılavuzu hazırlamak, yazmak ve yayımlamak görevi, Anayasa’nın 134. maddesine dayalı olarak çıkarılan 664 sayılı Kanun Hükmünde Kararname’nin 10. maddesinin ç fıkrasıyla Türk Dil Kurumuna verilmiştir. |
| 9 | Türk Dil Kurumu, kuruluşundan bu yana yazım kurallarının belirlenmesinde ve yazım kılavuzlarının yayımlanmasında kendisine düşen görevi yerine getirmeye çalışmaktadır. |
| 10 | Türk Dil Kurumu Güncel Türkçe Sözlük ve Yazım Kılavuzu Çalışma Grubu, yazımda yaşanan tartışmaları sona erdirmek ve herkesin benimseyeceği, kullanacağı yazım kurallarını ve yazım biçimlerini yaygınlaştırmak ilkesiyle her kuralın, her yazılışın üzerinde titizlikle durmuştur. |

Table 10: Testsuite English and Turkish for en-tr / tr-en.

| | | | | |
|---|---|---|---|---|
| en-tr | | | | |
| alibaba.5732 | Türk Dil Kurumu | 1.0 | 6 | 6 |
| | Türkçe Sözlü | 0.86 | 3 | 4 |
| | Yazım Kılavuzu Çalışma Grubu | 1.0 | 3 | 3 |
| | Yazım Kılavuzu | 0.6 | 3 | 7 |
| | yazım kural | 1.0 | 4 | 4 |
| | $F_1$ | 0.88 | 19 | 24 |
| alibaba.5744 | Türk Dil Kurumu | 1.0 | 6 | 6 |
| | Türkçe Sözlü | 0.86 | 3 | 4 |
| | Yazım Kılavuzu Çalışma Grubu | 1.0 | 3 | 3 |
| | Yazım Kılavuzu | 0.6 | 3 | 7 |
| | yazım kural | 1.0 | 4 | 4 |
| | $F_1$ | 0.88 | 19 | 24 |
| parfda | Türk Dil Kurumu | 1.0 | 6 | 6 |
| | Türkçe Sözlü | 1.0 | 4 | 4 |
| | $F_1$ | 0.59 | 10 | 24 |
| uedin.5644 | Türk Dil Kurumu | 1.0 | 6 | 6 |
| | yazım kural | 0.86 | 3 | 4 |
| | $F_1$ | 0.54 | 9 | 24 |
| online-B | Türk Dil Kurumu | 0.91 | 5 | 6 |
| | Türkçe Sözlü | 0.86 | 3 | 4 |
| | yazım kural | 0.4 | 1 | 4 |
| | $F_1$ | 0.54 | 9 | 24 |
| NICT.5695 | Türk Dil Kurumu | 1.0 | 6 | 6 |
| | yazım kural | 0.67 | 2 | 4 |
| | $F_1$ | 0.5 | 8 | 24 |
| RWTH.5632 | Türk Dil Kurumu | 1.0 | 6 | 6 |
| | $F_1$ | 0.4 | 6 | 24 |
| online-A | Türk Dil Kurumu | 0.8 | 4 | 6 |
| | $F_1$ | 0.29 | 4 | 24 |
| online-G | Türkçe Sözlü | 0.86 | 3 | 4 |
| | $F_1$ | 0.22 | 3 | 24 |
| tr-en | | | | |
| online-B | Turkish Language Institution | 0.5 | 2 | 6 |
| | Turkish Language | 0.86 | 8 | 6 |
| | Turkish Dictionary | 0.67 | 2 | 4 |
| | Working Group | 1.0 | 3 | 3 |
| | Writing Manual | 0.4 | 1 | 4 |
| | $F_1$ | 0.82 | 16 | 23 |
| NICT.5708 | Turkish Language Institution | 0.67 | 3 | 6 |
| | Turkish Language | 1.0 | 6 | 6 |
| | Working Group | 1.0 | 3 | 3 |
| | Writing Manual | 0.67 | 2 | 4 |
| | $F_1$ | 0.76 | 14 | 23 |
| uedin.5709 | Turkish Language Institution | 0.67 | 3 | 6 |
| | Turkish Language | 1.0 | 6 | 6 |
| | Working Group | 1.0 | 3 | 3 |
| | $F_1$ | 0.69 | 12 | 23 |
| parfda | Turkish Language Institution | 0.29 | 1 | 6 |
| | Turkish Language | 0.8 | 4 | 6 |
| | Working Group | 1.0 | 3 | 3 |
| | $F_1$ | 0.52 | 8 | 23 |
| online-G | Turkish Dictionary | 0.86 | 3 | 4 |
| | $F_1$ | 0.23 | 3 | 23 |
| online-A | Turkish Language | 0.29 | 1 | 6 |
| | $F_1$ | 0.08 | 1 | 23 |

Table 11: Testsuite $F_1$ scores with key phrases.

| | | BLEU lc | | | | BLEU | | | | $F_1$ lc | | | | $F_1$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | model | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| en-tr | alibaba.5732 | 0.4591 | 0.4482 | 0.3177 | 0.3112 | 0.2287 | 0.2255 | 0.1701 | 0.1684 | 0.3597 | 0.3537 | 0.2827 | 0.2789 | 0.2323 | 0.2294 | 0.1981 | 0.1958 |
| | **alibaba.5744** | 0.4778 | 0.4669 | 0.3297 | 0.3231 | 0.2394 | 0.2362 | 0.1785 | 0.1768 | 0.3717 | 0.3658 | 0.2936 | 0.2898 | 0.2416 | 0.2388 | 0.2055 | 0.2033 |
| | online-A | 0.3977 | 0.3612 | 0.2486 | 0.2027 | 0.1469 | 0.109 | 0.0714 | 0.0571 | 0.292 | 0.2532 | 0.2146 | 0.1829 | 0.1649 | 0.141 | 0.135 | 0.1156 |
| | online-B | 0.4412 | 0.423 | 0.3005 | 0.2831 | 0.2045 | 0.185 | 0.1296 | 0.1118 | 0.3419 | 0.3196 | 0.2631 | 0.2419 | 0.2083 | 0.1896 | 0.1711 | 0.1557 |
| | online-G | 0.4105 | 0.3778 | 0.2493 | 0.2172 | 0.1544 | 0.1308 | 0.0934 | 0.0824 | 0.3099 | 0.2788 | 0.2313 | 0.2064 | 0.1817 | 0.1629 | 0.15 | 0.1346 |
| | parfda | 0.3367 | 0.3258 | 0.1818 | 0.1718 | 0.1055 | 0.0934 | 0.0558 | 0.051 | 0.2417 | 0.2292 | 0.1775 | 0.1657 | 0.1375 | 0.1286 | 0.1123 | 0.105 |
| | uedin | 0.448 | 0.4334 | 0.2971 | 0.2808 | 0.191 | 0.1708 | 0.1231 | 0.1007 | 0.3428 | 0.3255 | 0.2572 | 0.2415 | 0.2033 | 0.189 | 0.1678 | 0.1551 |
| | NICT | 0.4376 | 0.4231 | 0.2851 | 0.2686 | 0.1947 | 0.1786 | 0.132 | 0.117 | 0.3404 | 0.3233 | 0.262 | 0.2472 | 0.2101 | 0.1972 | 0.1741 | 0.1629 |
| | RWTH | 0.3752 | 0.3643 | 0.2065 | 0.2001 | 0.119 | 0.1125 | 0.0676 | 0.0648 | 0.2507 | 0.2417 | 0.1803 | 0.1725 | 0.1395 | 0.1336 | 0.1142 | 0.1094 |
| tr-en | parfda | 0.413 | 0.4006 | 0.2122 | 0.1963 | 0.0963 | 0.0861 | 0.0 | 0.0 | 0.2586 | 0.2404 | 0.1782 | 0.1656 | 0.1343 | 0.1249 | 0.1086 | 0.1011 |
| | online-A | 0.5452 | 0.4877 | 0.3759 | 0.3056 | 0.2602 | 0.1988 | 0.1778 | 0.1369 | 0.4391 | 0.3573 | 0.3367 | 0.2661 | 0.2698 | 0.213 | 0.2253 | 0.1782 |
| | **online-B** | 0.5765 | 0.5656 | 0.4297 | 0.418 | 0.3244 | 0.3152 | 0.2459 | 0.239 | 0.4825 | 0.4732 | 0.3863 | 0.3786 | 0.3197 | 0.3135 | 0.2725 | 0.2679 |
| | online-G | 0.5414 | 0.4972 | 0.3843 | 0.3362 | 0.2803 | 0.2364 | 0.2038 | 0.168 | 0.4497 | 0.3929 | 0.353 | 0.3024 | 0.288 | 0.2441 | 0.243 | 0.2045 |
| | uedin | 0.5333 | 0.5227 | 0.3606 | 0.3385 | 0.2507 | 0.2222 | 0.1738 | 0.1426 | 0.424 | 0.4046 | 0.3253 | 0.3038 | 0.2603 | 0.2386 | 0.216 | 0.1961 |
| | NICT | 0.5339 | 0.526 | 0.3544 | 0.3478 | 0.2294 | 0.2244 | 0.1422 | 0.1399 | 0.4239 | 0.4199 | 0.3185 | 0.316 | 0.2502 | 0.2488 | 0.2064 | 0.2055 |

Table 12: Testsuite BLEU and $F_1$ results.

# The TALP-UPC Machine Translation Systems for WMT18 News Shared Translation Task

**Noe Casas, Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa**
contact@noecasas.com, carlos.escolano@tsc.upc.edu
{marta.ruiz,jose.fonollosa}@upc.edu
TALP Research Center
Universitat Politècnica de Catalunya, Barcelona

## Abstract

In this article we describe the TALP-UPC research group participation in the WMT18 news shared translation task for Finnish-English and Estonian-English within the multi-lingual subtrack. All of our primary submissions implement an attention-based Neural Machine Translation architecture. Given that Finnish and Estonian belong to the same language family and are similar, we use as training data the combination of the datasets of both language pairs to paliate the data scarceness of each individual pair. We also report the translation quality of systems trained on individual language pair data to serve as baseline and comparison reference.

## 1 Introduction

Neural Machine Translation (NMT) has consistently maintained state of the art results in the last years. However, due to its need for large amounts of training data, low resource language pairs need to resort to extra techniques to achieve acceptable translation quality.

In the WMT18 news shared translation task, two of the languages to translate are Finnish and Estonian (that are to be translated to and from English). Both can be considered low-resource languages in general, and also in particular for this shared task, based on the volume of data made available for training, especially Estonian.

In this report we describe the participation of the TALP research group from *Universitat Politècnica de Catalunya* (UPC) at the aforementioned WMT18 news shared translation task, specifically in the multi-lingual subtrack, as our systems make use of the data from both Finnish and Estonian language to improve the translation quality.

## 2 Linguistic Background

Finnish and Estonian are respectively the official languages of Finland and Estonia, having 5.4 and 1.1 million native speakers (Lewis, 2009). They are **Finnic Languages**, a branch within the Uralic Language family.

Estonian and Finnish make use of the Latin alphabet with some additional letters, each one incorporating extra letters (e.g. ä, ö, ü, õ, š, ž).

Finnish and Estonian are morphologically-rich **agglutinative languages**. Estonian presents fourteen grammatical cases while Finnish presents fifteen. Verb conjungations are very regular in both languages. Neither of them has grammatical gender nor definite or indefinite articles. Both have flexible word order, but the basic order is subject-verb-object.

Like other Finnic languages, both Finnish and Estonian present consonant gradation (consonants are classified in grades according to phonologic criteria, and such grades condition the combined appearance of the consonants in a derived word), but the gradation patterns each one follows are different.

While Finnish has kept most of its late Proto-Finnic linguistic traits, Estonian has lost some of its former characteristics, like vowel harmony (vowels in a word cannot appear freely but their allowance is constrained by rules), which in Finnish affects case and derivational endings. Also, Estonian mostly lost the word-final sound, making its inflectional morphology more fusional for nouns and adjectives (Fortescue et al., 2017). German language influence also led Estonian to use more postpositions where Finnish uses cases. Geographical location has also led to differences in the loanwords borrowed by each language.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 355–360
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64033

## 3 Attention-based NMT

The first competitive NMT systems were based on the sequence-to-sequence architecture (Cho et al., 2014; Sutskever et al., 2014), especially with the addition of attention mechanisms (Bahdanau et al., 2014; Luong et al., 2015), either using Gated Recurrent Units (GRU) (Cho et al., 2014) or Long-Short Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997).

Sequence-to-sequence with attention was the state of the art NMT model until the Transformer architecture (Vaswani et al., 2017) was proposed. This model does not rely on recurrent units or convolutional networks, but only on attention layers, combining them with several other architectural elements: positional embeddings (Gehring et al., 2017), layer normalization (Ba et al., 2016), residual connections (He et al., 2016) and dropout (Srivastava et al., 2014).

The type of attention mechanism used by the Transformer model is a multi-headed version of the dot-product attention, applied both as self-attention to source and target (prefix) sentences and as encoder-decoder attention mechanism.

## 4 Low resource NMT

The application of NMT to low resource language pairs needs extra techniques to achieve good translation quality. These are some of the frequently used approaches:

**Back-translation** (Sennrich et al., 2015a) consists in training an auxiliary translation system from target language to source language and use it to translate a large target language monolingual corpus into the source language, and then use such synthetic source-target sentence pairs to augment the originally available parallel corpus and train on it a new source language to target language translation system.

**Pivoting approaches** use a third resource-rich language as *pivot* and train translation systems from source language to pivot and from pivot to target language. These auxiliary systems can either be used in *cascade* to obtain source-to-target translations, or be used to build synthetic parallel source-target corpora (i.e. *pseudocorpus approach*). A recent application of pivoting techniques to NMT can be found in (Costa-jussà et al., 2018).

**Adversarial learning** (Lample et al., 2018; Artetxe et al., 2018) in a multi-task learning setup, with an auxiliary text (denoising) auto-encoding loss, whose internal sentence representation is aligned with the ones from the translation task by means of a discriminator in feature space.

**Pre-trained cross-lingual embeddings** (Artetxe et al., 2016, 2017) can be used complementarily to further reduce the need for parallel data.

**Finding parallel data from a similar source language** and the same target language (or vice versa) and adding it to the original parallel corpus. With such a composite training data set, a wordpiece-level vocabulary can leverage the common word stems between the similar languages and profit from the combined amount of data. This approach is used in the present work, as described in sections 5 and 6.1.

**Multilingual zero-shot translation** (Johnson et al., 2017) also uses parallel corpora from different source and target language pairs, but mixes together every available language pair, regardless of how linguistically close they are. This way, there is a single shared word-piece vocabulary for all languages, and the system is trained on a corpus that combines data from several different language pairs. In order to convey the association between a source sentence and its translation to a specific target language, the source sentence is prefixed with a token that specifies which language the target sentence belongs to. This approach aims at implicitly learning language-independent internal representations, enabling the translation of low resource language pairs (and even language pairs where there is zero parallel data available) to profit from the combined language pair training data.

## 5 Corpora and Data preparation

All proposed systems in this work are constrained using exclusively parallel data provided by the organization. For the English - Finnish language pair the data employed is the Europarl corpus version 7 and 8, Paracrawl corpus, Rapid corpus of EU press releases and Wiki Headlines corpus. For the English - Estonian data the Europarl v8 corpus, Paracrawl and Rapid corpus of EU press releases corpus were employed.

All language pairs have been preprocessed following the proposed scripts by the organization of the conference. The pipeline consisted in normalizing punctuation, tokenization and truecasing using the standard *Moses* (Koehn et al., 2007) scripts. With the addition that, for tokenization, no escaping of special characters was performed.

For the language pair of English - Estonian we found that from Paracrawl corpus a considerable number of sentences were not suitable sentences in the intended languages, but apparently random sequences of upper case characters. In order to remove them, an additional step of language detection was performed using library `langdetect` (Danilák, 2017), which is a port to Python of library `language-detection` (Shuyo, 2010). The criteria for removing noisy sentences from the dataset was that either one of the languages of the pair could not be identified as a language.

The sizes of the different data sets compiled for each language pair and once cleaned as described earlier in this section are presented in Table 1.

Table 1: Corpus statistics in number of sentences and words for both parallel corpora, English - Estonian and English - Finnish.

| corpus | lang | set | sentences | words |
|---|---|---|---|---|
| *En-Et* | *En* | train | 998547 | 23056922 |
| | | test | 2000 | 44305 |
| | *Et* | train | 998547 | 17376004 |
| | | test | 2000 | 34733 |
| *En-Fi* | *En* | train | 3064124 | 62208347 |
| | | dev | 3000 | 64611 |
| | | test | 3002 | 63417 |
| | *Fi* | train | 3064124 | 45692989 |
| | | dev | 3000 | 48839 |
| | | test | 3002 | 46572 |

As described in sections 2 and 4, as Finnish and Estonian belong to the Finnic language family and are similar to each other, we aimed at combining the individual parallel corpora (*En* - *Fi* and *En* - *Es*) into a single larger corpus. For the translation directions where English is the target language (i.e. *Fi* → *En* and *Et* → *En*) we prepared a combined *Fi* + *Et* → *En* corpus by simply concatenating the original ones. This approach was not applicable to the reverse directions, as we needed some way to convey the information about whether to generate either Finnish or Estonian as part of the input to the neural network. Following the approach in (Johnson et al., 2017), we modify the individual parallel corpora to add a prefix to the English sentences to mark whether the associated target sentence was Finnish or Estonian, and then proceed to concatenate both corpora into the final combined one *En* → *Fi* + *Et*. The prefixes used were respectively `<fi>` and `<et>`. This prefix needs to be added likewise to the test English sentences when decoding them into Finnish or Estonian.

As the combined corpora are concatenations of the individual ones, their sizes can be computed from the figures in Table 1 by mere addition of the individual sizes of each language pair.

## 6 System Description

In this section we present the translation systems used for our submissions, both in terms of vocabulary extraction strategies followed (section 6.1), of neural architecture used (section 6.2) and of needed post-processing (section 6.3).

### 6.1 Vocabulary Extraction

The NMT models used for all of our submissions, which are described in section 6.2 make use of predefined sets of discrete tokens that comprise the *vocabulary*.

The vocabulary of each of our translation systems (both the final submissions and the systems trained for reference described in section 7) was based on wordpiece extraction (Wu et al., 2016). For each system, the source and target vocabularies were extracted separately, aiming at a vocabulary size of 32K tokens. Vocabularies are not shared between source and target languages in any case.

Word-piece vocabularies (or the very similar Byte-Pair Encoding (BPE) vocabularies (Sennrich et al., 2015b)) are usually applied to extract vocabularies from corpora that contain data from similar languages in order to try to find common stems and derivational suffixes so that the language commonalities can be leveraged by the neural network training.

### 6.2 NMT Models

All the submissions presented to the task make use of the Transformer NMT architecture, which is described in section 3. We used the implementation released by the authors of (Vaswani et al., 2017) [1]

The complete hyperparameter configuration used for all the attention-based neural machine

[1] The authors of (Vaswani et al., 2017) made the source code available at `https://github.com/tensorflow/tensor2tensor`. For this work, version `1.2.9` was used.

translation models in our submissions (which consisted in the `transformer_base` parameter set in `tensor2tensor`) is shown in Table 2.

Table 2: Hyperparameters of the neural model.

| hyperparameter | value |
|---|---|
| attention layers | 6 |
| attention heads per layer | 8 |
| hidden size (embedding) | 512 |
| batch size (in tokens) | 4096 (4 GPU) |
| training steps | 800000 |
| tokenization strategy | wordpiece |
| vocabulary size | 32K |
| optimization algorithm | Adam |
| learning rate | warmup + decay |

After the training, the weights of the last 5 checkpoints (having checkpoints stored every 2000 optimization steps) are averaged to obtain the final model.

### 6.3 Post-processing

Following the inverse steps of the processing described in section 5, the decoded outputs of NMT model need to be de-truecased and de-tokenized by means of the appropriate *Moses* scripts.

## 7 Experiments

The hypothesis on which we base this work is that, given the similarity between Estonian and Finnish, a system trained with the combination of the data from both languages would outperform systems trained on the individual language datasets.

In order to validate this hypothesis, we conducted direct experiments, training systems on the individual language datasets and also on the combined datasets (as described in section 5), and comparing their translation quality. The datasets used for testing the performance were `newsdev2018` for Estonian - English and `newstest2017` for Finnish - English. The results of the experiments are shown in Table 3, were all figures represent case-insensitive BLEU score over the aforementioned reference test corpora.

While the results for Finnish are not very different between the individual and combined data trainings [2], the results for Estonian show an important improvement of the training on the combined data over the individual data. This correlates with the fact that the Estonian - English training set is less than one third the size of the Finnish - English, therefore the size increase in the Finnish - English combined training corpus is much smaller than the increase for Estonian - English, as shown in Table 1.

Table 3: Comparison between translation quality (case-insensitive BLEU) of systems trained on the individual language data vs. systems trained on the combined data .

| direction | individual | combined | ΔBLEU |
|---|---|---|---|
| *En* → *Fi* | 24.36 | 25.21 | +0.85 |
| *Fi* → *En* | 29.39 | 30.00 | +0.61 |
| *En* → *Et* | 15.97 | 18.92 | +2.95 |
| *Et* → *En* | 21.66 | 25.66 | +4.00 |

## 8 Conclusions

In this article we described the TALP-UPC submissions to the the multi-lingual subtrack of the WMT18 news shared translation task for Finnish - English and Estonian - English language pairs.

Our experiments suggest that for low resource languages, enlarging the training data with translations from a similar language can lead to important improvements in the translation quality when using subword-level vocabulary extraction strategies. In this line, further research should be conducted to understand how subwords have captured the differences between Estonian and Finnish cognates and to leverage such an insight to devise more effective vocabulary extraction strategies.

## Acknowledgements

We would like to thank Magdalena Biesialska for her help in the news task human evaluation.

This work is partially supported by Lucy Software / United Language Group (ULG) and the Catalan Agency for Management of University and Research Grants (AGAUR) through an Industrial PhD Grant. This work is also supported in part by the Spanish Ministerio de Economía y Competitividad, the European Regional Development Fund and the Agencia Estatal de Investigación, through the postdoctoral senior grant Ramón y Cajal, contract TEC2015-69266-P (MINECO/FEDER,EU) and contract PCIN-2017-079 (AEI/MINECO).

[2] Improvements of less than 1 BLEU point are normally considered neglectable.

## References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 451–462.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. 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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734.

Marta R. Costa-jussà, Noe Casas, and Maite Melero. 2018. English-catalan neural machine translation in the biomedical domain through the cascade approach. In *Proceedings of the 11th Language Resources and Evaluation Conference of the European Language Resources Association*.

Michal Danilák. 2017. Langdetect. https://github.com/Mimino666/langdetect.

Michael Fortescue, Marianne Mithun, and Nicholas Evans. 2017. *The Oxford Handbook of Polysynthesis*. Oxford Handbooks. Oxford University Press.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

M. Paul Lewis, editor. 2009. *Ethnologue: Languages of the World*, sixteenth edition. SIL International, Dallas, TX, USA.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Nakatani Shuyo. 2010. Language detection library for java.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

# Phrase-based Unsupervised Machine Translation with Compositional Phrase Embeddings

**Maksym Del** **Andre Tättar** **Mark Fishel**
Institute of Computer Science
University of Tartu, Estonia
{maksym.del,andre.tattar,fishel}@ut.ee

## Abstract

This paper describes the University of Tartu's submission to the unsupervised machine translation track of WMT18 news translation shared task. We build several baseline translation systems for both directions of the English-Estonian language pair using monolingual data only; the systems belong to the phrase-based unsupervised machine translation paradigm where we experimented with phrase lengths of up to 3. As a main contribution, we performed a set of standalone experiments with compositional phrase embeddings as a substitute for phrases as individual vocabulary entries. Results show that reasonable n-gram vectors can be obtained by simply summing up individual word vectors which retains or improves the performance of phrase-based unsupervised machine tranlation systems while avoiding limitations of atomic phrase vectors.

## 1 Introduction

Most successful approaches to machine translation (Wu et al., 2016; Bahdanau et al., 2014; Vaswani et al., 2018; Gehring et al., 2017) rely on the availability of parallel corpora. Supervised neural machine translation (NMT) employs the encoder-decoder architecture, where the encoder reads the source sentence and produces its representation which is then fed to the decoder that tries to generate the target sentence word by word. Cross-entropy loss is usually used as a training objective and beam search algorithm is used for inference. These neural models show state-of-the art performance but rely on vast amounts of parallel data.

On the other hand, there is a Statistical Machine Translation paradigm that is based on phrase tables that are learned from parallel corpus. These methods are currently replaced by neural counterparts for high-resource languages, but perform better in low-resource settings (Bentivogli et al., 2016).

For some language pairs the size of parallel corpus ranges from extremely low to almost zero. These extremely low-resource language pairs were a motivation for the Unsupervised Machine Translation (Lample et al., 2017; Artetxe et al., 2017b) that aims to translate language without usage of the parallel corpora for training.

The first step of the unsupervised approach to translation is the same for all methods: learning word level embedding spaces for source and target languages and then aligning these spaces. Next, one of the unsupervised vector space mapping methods (Artetxe et al., 2017a; Conneau et al., 2017) is applied to align spaces together and perform word by word translation. This mapping is only possible because of the linear properties of the word embedding method that is used to get word vector representations. Lastly, to improve system's performance, iterative refining using either neural network models or parts of SMT pipeline is done (Mikolov et al., 2013b; Lample et al., 2018). In this work we implement a system that is similar to the latter approach.

Statistical Machine Translation systems that are based on the phrase representations require smaller amounts of data to achieve reasonable performance. Phrase-based Unsupervised Machine Translation is motivated by the assumption that methods working without parallel data can benefit from the usage of phrases as the basic units. In order for phrases to be used for Unsupervised Translation they have to be represented in a suitable way. Current approach is to learn phrase embeddings as atomic vocabulary units (Lample et al., 2018).

In this work, we implement systems for the English-Estonian language pair following the guidlines presented in (Lample et al., 2018) and

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 361–367
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64034

show that simply using sum of individual word embeddings can produce reasonable phrase embeddings. This eliminates the need of having huge n-gram vocabulary that might be hard to learn and use in unsupervised bilingual mapping. It also allows to obtain embeddings for arbitrary phrases as opposite to using predefined limited set of phrases.

This paper is organized as follows: In Section 2 we describe our unsupervised translation system baseline. Section 3 describes our approach to computing phrase embeddings for arbitrary phrases for UMT. In Section 4 we describe our experiments for compositional phrase embeddings standalone and in context of UMT. Section 5 concludes the work.

## 2 Baseline UMT System

Our baseline systems relies on the recently proposed Phrase-based UMT framework (Lample et al., 2018). Firstly, we learn monolingual embeddings for words and then use unsupervised mapping for bilingual lexicon extraction. The lexicon is used to compute semantic distances between phrases which results in the phrase table as used in Statistical Machine Translation. Lastly, we use standard SMT pipeline [1] with ngram language models in order to generate translations. We do not perform iterative back translation for simplicity and due to time and compute limits.

In the simplest case the phrase table consists solely of the unigram entries, but following Lample et al. we also consider bigram and trigram expermients. However, we use a different procedure for making decisions on which ngrams to consider for adding to the phrase table.

Extraction of n-grams is done by probabilistically joining words into phrases. We use frequency filtering with sampling, so n-grams are sometimes joined and sometimes not. We downsample the most frequent words, with probability function $p = \frac{1}{f^{\beta}}$, where p is the sampling probability, f is the n-gram frequency and B is a small weight (we used $\beta = \frac{1}{8}$). Frequency filtering is used for very rare words, so they must appear more than the set threshold. For example a n-gram that appears 25 times is joined with probability 0.668, but n-gram that appears 1000 times is joined with probability 0.422.

The point of using phrases is that they should be less ambiguous than words, and the fact that one word in one language can be a phrase in another language, like Estonian word ”laualt”, which means ”from the table”. The extraction of n-grams is done as in Blue2vec algorithm (Tättar and Fishel, 2017). To compute embeddings (S. Harris, 1954; Mikolov et al., 2013b) we use FastText[2] (Bojanowski et al., 2017) embeddings instead of word2vec (Mikolov et al., 2013a). We prefer FastText because it produces embeddings that incorporate subword level information which is proven to be helpful.

After finding vectors for words and phrases, we need to project the source and target language embeddings into the same space (Artetxe et al., 2017a; Conneau et al., 2017; Artetxe et al., 2018). Projecting is done using the MUSE [3] library. We use cross lingual similarity scores to score N-grams.

## 3 Approach to Phrase Embeddings

State of the art Phrase Based UMT systems ( Lample et al.) learn phrase embeddings as individual vocabulary entries (to form the entry, we just concatenate words together with underscore). Although this approach provides good embeddings for phrases, it has serious limitations. Firstly, it is very memory intensive because it is infeasible to learn and store vocabulary of all phrases that can occur in the corpus. The size of the vocabulary grows almost exponentially over the length of the phrase, and thus vocabularies of that order of magnitude do not fit into the computer memory in most cases. Secondly, there is a data sparsity problem since some (even two-word) phrases occur rarely even in the very large corpus (it makes learning embedding vectors hard for some phrases).

The idea to combine embeddings to get a phrase embedding (Mikolov et al., 2013b) was successfully used in context of tasks such phrase similarity (Muraoka et al., 2014) and non-compositional phrase detection (Yazdani et al., 2015). We follow similar idea and show why it is highly suitable specifically for unsupervised translation.

In summary, we first learn vectors for reasonable amount of phrases as single-token vocabulary entries (e.g. ”research_paper”). At the same time (as a part of the same training procedure), we learn embeddings for individual words (”research” and ”paper” separately). Finally, we train a regression

[1] http://www.statmt.org/moses/

[2] https://github.com/facebookresearch/fastText

[3] https://github.com/facebookresearch/MUSE

model to predict phrase vectors from their word vectors. We assume that the model will learn the function that captures the pattern of combining word vectors in order to generate phrase vectors. Since we can obtain vectors for arbitrary words, we thus can estimate vectors for arbitrary phrases by combining word vectors with the learned function. Remaining text of the subsection defines the pipeline in step by step fashion.

**Step 1: obtain training data for non-compositional modeling**. In order to compute vectors for words and subset of phrases (we treat phrases as single-unit vocabulary entries at this point) we first need to get a monolingual corpus and do some preprocessing like lowercasing / truecasing and tokenization.

**Step 2: extract phrase candidates**. Since we can not learn vectors for all phrases in the corpus, we have to decide which phrases to use to learn vectors for. One option is to randomly glue desired number of phrases together while other options include only gluing phrases that belong to some specific set of desired phrases. The set can be formed by scoring all phrases from the corpus by some criterion and then taking top N phrases based on their scores. Here we provide non-comprehensive list of criterion that can serve to the purpose of gluing two-word phrases: Likelihood ratio, Raw frequency, Poisson Stirling criterion, Chi square score, Dice score, Jaccard measure etc. We refer author to external literature for additional information on this topic (e.g. (Manning et al., 2008)). Concrete metric should be task specific or empirically chosen.

**Step 3: glue phrases**. At this step we simply go through the corpus from Step 1 and glue some words together based on the phrases set from the Step 2.

**Step 4: train word embedding model**. At this step we train (say) the Skip-gram model on words and phrases corpus from Step 3. This way we get semantic vectors for words and some phrases.

**Step 4: obtain training data for compositional modeling**. At this step we extract phrase vectors for phrases that we glued at the Step 3. Then we extract word vectors for words that are used to compose these phrases. The dataset then consists of following pairs of entities: sequence of the word vectors as an input, and phrase vector as the target.

**Step 5: train compositional model on the data from the Step 4**. At this point we use the dataset from previous step to teach the model to compose word vectors in a way that phrase vector is produced

One critical property of this framework is that it produces vectors for phrases as if they were learned as the single-token units as a part of the vocabulary. That is, result phrase embeddings will not differ (in terms of their properties) from word embeddings. Word embeddings are learned as individual vocabulary units and satisfy to all assumptions of bilingual mapping methods (Artetxe et al., 2018); therefore, phrase embeddings learned this way would also do. Since billingual embedding mapping is the key necessury step of all current approaches to UMT, our phrase embeddings might become strong alternative for any existing UMT system.

We make our implementation of the pipeline available as an open source project [4].

## 4 Experiments

### 4.1 Compositional Phrase Embeddings

In this subsection we describe our experiments on compositional phrase embeddings standalone.

#### 4.1.1 Setup

We explored on the predictive ability of the different variants of the compositional models that we train as a part of the Step 5 of the framework. Following steps describe concrete decisions we made as a part of our implementation of the general pipeline we defined in previous subsection.

**Step 1: obtain training data for non-compositional modeling**. We used first 1 billion bytes of English Wikipedia as our training data. The data contains 124,301,826 lowercased tokens.

**Step 2: extract phrase candidates**. We only glued phrases that belong to some specific set of desired phrases. The set was formed by scoring all phrases from the corpus by likelihood ratio criterion and then taking top 600,000 phrases based on their scores.

**Step 3: glue phrases**. At this step we simply went through the corpus from Step 1 and glued some words together based on the phrases set from the Step 2.

**Step 4: apply skip-gram model**. At this step we trained Skip-gram model on words and phrases

[4] https://github.com/maxdel/bigram_embedder

corpus from Step 3. This way we got semantic real valued vectors for words and some phrases. We trained the system using *fasttext* framework ( (Bojanowski et al., 2017)) for 6 epochs with default parameters expect for the embedding size which we set to 100.

**Step 4: obtain training data for compositional modeling**. The result dataset size was about 600,000 training examples.

**Step 5: train compositional model on the data from the Step 4**. At this point we used the dataset from previous step to teach different models to compose word vectors in a way that phrase vector is produced.

We also left some examples apart for validation and testing. The test set size was 2400 examples, development test size was 2000 examples. Development set was used to tune models hyperparameters, and test set was used to perform final models comparison.

### 4.1.2 Candidate Models

Let *w1* be the vector of the first word and *w2* the vector of the second. Let also *p* be the result vector of the phrase and D be the dimension of the w1, w2, and p. The types of the models that we implemented and trained[5] are following.

- Simple addition *(AddSimple)*:

$$p = w1 + w2$$

- Addition with attention weights *(AddAtt)*:

$$p = a1 * w1 + a2 * w2$$

where a1 and a2 are scalars that are learned by first concatenating the word vectors, and then projecting result into two dimensions.

- Dimwise addition with attention weights *(AddAttDimwise)*:

$$p = a1 * w1 + a2 * w2$$

where a1 and a2 are vectors of the size D that are learned by first concatenating the word vectors, and then projecting result into the D dimensions. Therefore, we add two vectors with weight assigned to each dimension.

- Neural Network with one linear layer *(Linear)*:

$$W1 * ([w1, w2])$$

where W1 is parameters matrix and [w1, w2] means concatenation

- Neural Network with dense ReLU layer and linear layer *(NonLinear)*:

$$W2 * ReLU(W1 * [w1, w2])$$

- Multilayer Neural Network *(MultilNonLinear)*: the same as the previous one, but with two more nonlinear layers. The sizes of hidden layers are 170, 130, and 100.

- Long Short Term Memory network *(LSTM)*: last timestep is used as phrase representation ( (Hochreiter and Schmidhuber, 1997)).

*Smooth l1 loss* was used in order to train all the models:

$$\text{loss}(x, y) = \frac{1}{n}\sum_i z_i,$$

where

$$z_i = \left\{ \begin{array}{lc} 0.5(x_i - y_i)^2, & \text{if}|x_i - y_i| < 1 \\ |x_i - y_i| - 0.5, & \text{otherwise} \end{array} \right\}$$

We choose this loss because it is more tolerant to outliers, which may occur due to non-compositionality of some phrases.

### 4.1.3 Experiments Results and Analysis

In order to get interpretable accuracy scores for models comparison we first run our trained regression models in the inference mode to predict phrase vectors for the test set. Then we retrieve the top N (N is one of {1,3,5,10}) closest points in the embedding space, and check if the ground truth phrase belongs to this set. If the phrase is not in the topN, we count it as an error. Lastly we divide number of non-error examples by the size of the training set to obtain accuracy score. Table 1 shows accuracy scores across various models across various topN values.

As we can see from the Table 1, simple summation baseline shows decent performance in compositional phrase vector modeling. This interesting result was explained by the authors of the Skip-gram model ( (Mikolov et al., 2013b)). They

[5] *AddSimple* model does not require training while attention weights for *AddAtt* and *AddAttDimwise* are learned from data

Table 1: Accuracy results of explicit evaluation of compositional models. Top 3 results among columns are in bold.

| **Model** | **top1** | **top3** | **top5** | **top10** |
|---|---|---|---|---|
| AddSimple | ***0.35*** | **0.81** | **0.88** | **0.94** |
| AddAtt | 0.37 | 0.65 | 0.74 | 0.84 |
| AddAttDimwise | 0.38 | 0.66 | 0.75 | 0.84 |
| Linear | **0.71** | **0.85** | **0.88** | **0.92** |
| NonLinear | 0.62 | 0.75 | 0.80 | 0.85 |
| MultiNonLinear | **0.69** | 0.83 | 0.87 | 0.91 |
| LSTM | **0.73** | **0.88** | **0.92** | **0.95** |

show that addition of two token vectors approximately equivalent to the AND operation between their distributions over context words (we predict context / surrounding words with the Skip-gram model). This means that the result token vector will be equivalent to the token (phrase or word) that shares the same context with the input token vectors. Despite the good performance of the simple addition function, we observe drop in performance for attentional analogues. This might be due to the fact that it is sometimes hard to predict these attention weights from the words themselves, since the Skip-gram embeddings does not really contain much of Part of Speech (POS) information (e.g. words like ”go”, ”goes”, ”going”, ”went” are grouped together despite having different POS tags) while this information is what was needed to achieve good results in (Muraoka et al., 2014). Among neural architectures, the LSTM network was able to outperform simple sum operation. It might be due to the separate gates it uses for memorizing the important (in context of the future phrase) semantic part of the word, forgetting redundant dimensions, and updating the first word with some information from the second word. We conducted experiments on phrases with length up to two words for simplicity.

This experiments show that LSTM network is a powerful tool for predicting compositional phrases while linear layer remains a strong option. However, we also strongly consider summation as a valid option due to its simplicity, comparable performance, and theoretical motivation. In fact, we use summation powered phrase embeddings in our ongoing experiments with phrase-based UMT system described here.

Note that sum shows low performance at top1 sampling (Table 1). It is explainable since there is no information about the order in which individual words were summed so model just outputs the vector for a phrase with reverse word order. However, it is still acceptable vector as it is shown by a huge jump at top3, where usually both word ordering options are included.

### 4.2 Unsupervised Machine Translation

In this subsection we describe our experiments with compositional phrase embeddings as a part of the phrase based UMT system.

#### 4.2.1 Setup

In this subsection, we describe the the setupparameters we used in our system. The UnsupervisedMT framework [6] was used to train baselines.

**Data.** In our experiments we use datasets from the WMT’18 unsupervised translation task (Bojar et al., 2018) for Engish-Estonian language pair. 15M monolingual sentences for Estonian and the same amount for English is considered.

**Preprocessing.** We use tokenization and truecasing to preprocess our data for both embeddings learning and language models training.

**Unigram system.** This system contains unigram cross-lingual word embeddings and does not include the n-grams.

**Bigram-atomic/trigram-atomic system.** This systems use phrase table that consist of bigram/trigram entries. Embedding vectors for bigrams/trigrams were obtained as a result of treating ngrams as atomic vocabulary units.

**Bigram/trigram-sum.** This system uses phrase table that consist also of bigram/trigram entries, embedding vectors for which were obtained as a result of summing individual word vectors that form the bigram. The proportion of unigram / bigram / trigram types is about 0.5 / 0.25 / 0.25 for English and 0.89 / 0.9 / 0.2 for Estonian.

**N-gram extraction.** Frequency filter settings are the following: 20, 120 and 90 (frequency counts for filtering out infrequent n-grams, unigrams, bigrams and trigrams respectively). The beta parameter is set to 0.125. This procedure is used to extract ngrams for both the atomic and sum systems. As a result, for bigram experiments, 38% of English vocabulary and 9% of estonian vocabularty consisted of bigrams.

**FastText Embeddings.** We use CBOW with character n-grams of size 3 to 6 as a core algo-

[6] https://github.com/facebookresearch/UnsupervisedMT

rithm for embeddings learning. The number of dimensions is set to 300. Other parameters are kept as default.

**Bilingual MUSE embeddings.** Default parameters are kept and MUSE is used on CUDA GPUs. Embeddings dimensions are set to 300.

**Language model**. We train Moses style ngram langage models with the order of 5.

**Moses and beam search**. We keep all Moses and beam search hyperparameters default.

### 4.2.2 Experiments Results and Analysis

Final BLEU scores for the experiments with Unigram, Bigram/Trigram-atomic, and Bigram/Trigram-sum systems are presented in Table 2.

Table 2: BLEU scores for our baselines and systems powered with compositional phrase embeddings for Estonian-English and English-Estonian language directions.

| System | et-en | en-et |
|---|---|---|
| Unigram | 4.42 | 4.14 |
| Bigram-atomic | 7.92 | 4.73 |
| Trigram-atomic | 7.63 | 4.96 |
| Bigram-sum | 6.25 | 3.88 |
| Trigram-sum | 6.28 | 4.05 |

Regarding Ngram-atomic series of experiments, Table 2 shows that there is an advantage of using bigrams for both language directions. However, for Estonian-English the advantage is much bigger (+3.5 against +0.59 BLEU). That might be due to the fact the the number of ngrams in English vocabulary is more then 3 times as big as the corresponding number for Estonian vocabulary. Trigram experiment provides no significant increase or decrease over the bigram experiment.

The benefit of using bigrams and the trigram trend are consistent with the findings of Lample et al.. However, while Lample et al. reports the increase of about 1 BLEU point when using bigrams, we observe the increase of 3.5 for et-en. Note that we also use custom ngram extraction procedure as opposite to taking top N most frequent bigrams (Lample et al., 2018).

In case of Ngram-sum experiment for et-en, the system outperforms unigram experiment suggesting that the compositional embeddings do have semantic power. However, it is below the Bigram-atomic and Trigram-atomic baselines which is expected since the predictivness of the compositional model is not perfect. For en-et however, neither Bigram-sum nor Trigram-sum system outperforms atomic baseline suggesting that the topic needs additional dedicated research efforts.

## 5 Conclusions and Future Work

In this work, we present our results for the WMT18 shared task on unsupervised translation. Our baseline systems follow principles of the Phrase-based Unsupervised MT where we study unigram, bigram, and trigram systems. The vectors for ngrams are learned as individual vocabulary entries which has its limitations. Thus we study compositional phrase embeddings as a substitute, and show that simply summing up individual phrase words results in phrase embeddings that allow UMT systems to improve over baselines.

We showed that atomic phrase embeddings can be accurately estimated with compostional predictive models. Still, the effect of compositional phrase embeddings on PBUMT is still to be studied. More language pairs should be considered and more exhaustive targeted experiments with stronger baselines should be done. We leave this research direction for future work.

## Acknowledgments

This work was supported by the Estonian Research Council grant no. 1226.

The authors would like to thank the University of Tartu's Institute of Electronics and Computer Science for providing GPU computing resources.

## References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017a. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5012–5019.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017b. Unsupervised neural machine translation. *CoRR*, abs/1710.11041.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. *CoRR*, abs/1608.04631.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of WMT'18: the Third Conference on Machine Translation*, Brussels, Belgium.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *CoRR*, abs/1710.04087.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *CoRR*, abs/1804.07755.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Masayasu Muraoka, Sonse Shimaoka, Kazeto Yamamoto, Yotaro Watanabe, Naoaki Okazaki, and Kentaro Inui. 2014. Finding the best model among representative compositional models. In *PACLIC*.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10:146–162.

Andre Tättar and Mark Fishel. 2017. bleu2vec: the Painfully Familiar Metric on Continuous Vector Space Steroids. pages 619–622.

A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. N. Gomez, S. Gouws, L. Jones, Ł. Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer, and J. Uszkoreit. 2018. Tensor2Tensor for Neural Machine Translation. *ArXiv e-prints*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Majid Yazdani, Meghdad Farahmand, and James Henderson. 2015. Learning semantic composition to detect non-compositionality of multiword expressions. In *EMNLP*.

# Alibaba's Neural Machine Translation Systems for WMT18

**Yongchao Deng*** **Shanbo Cheng*** **Jun Lu*** **Kai Song*** **Jingang Wang*** **Shenglan Wu***
**Liang Yao*** **Guchun Zhang*** **Haibo Zhang*** **Pei Zhang*** **Changfeng Zhu***
**Boxing Chen**

Machine Intelligence Technology Lab, Alibaba Group
{yongchao.dyc,shanbo.csb,jeolu.luj,songkai.sk,jingang.wjg,shenglan.wsl,
yaoliang.yl,guchun.zgc,zhanhui.zhb,xiaoyi.zp,changfeng.zcf,
boxing.cbx}@alibaba-inc.com

## Abstract

This paper describes the submission systems of Alibaba for WMT18 shared news translation task. We participated in 5 translation directions including English ↔ Russian, English ↔ Turkish in both directions and English → Chinese. Our systems are based on Google's Transformer model architecture, into which we integrated the most recent features from the academic research. We also employed most techniques that have been proven effective during the past WMT years, such as BPE, back translation, data selection, model ensembling and reranking, at industrial scale. For some morphologically-rich languages, we also incorporated linguistic knowledge into our neural network. For the translation tasks in which we have participated, our resulting systems achieved the best case sensitive BLEU score in all 5 directions. Notably, our English → Russian system outperformed the second reranked system by 5 BLEU score.

## 1 Introduction

We participated in the WMT18 shared news translation task in 3 different language pairs: English ↔ Russian, English ↔ Turkish and English → Chinese. English ↔ Russian is a traditional WMT language pair possessing a large amount of bilingual training and development data. And especially this year, 16 million new translation units are available for the training. However for some more recent language pairs, the situation of bilingual resources is less promising: English ↔ Turkish language pair only has 200 K bitexts and for English → Chinese, the amount of bilingual resources remains the same as last year. In the following sections of this article, We will see that the availability of bilingual resources can differentiate the performance of the final system. More precisely, more bilingual data means greater ability to interact and absorb target side monolingual knowledge through the process of back translation, as well as its ability to retrieve the pertinent in-domain data during the data selection process.

We share a very similar model architecture and training flow for different languages directions. Our models are based on the Google's Transformer architecture (Vaswani et al., 2017). In order to improve our single system's performance, we experiment with some latest research findings such as transformer with relative position attention (Shaw et al., 2018), weighted transformer (Ahmed et al., 2017) and neural suffix prediction for Russian (Song et al., 2018) which will be developed in the next section. We will also see that different well-known multi-system based techniques such as model ensembling and model reranking can still improve the performance of a very strong single system, even though we have to push further the limit in term of the number of models to employ as well as the methods to combine them together.

The paper is structured as follows: Section 2 will describe the novelties of our model architecture compared to the Google's standard Transformer framework, then we present a detailed overview of our system in Section 3, before giving the experimental settings and main results across languages in Section 4. Finally, Section 5 will draw a brief conclusion of our work for WMT18.

## 2 Model Features

We describe in this section three different architecture enhancements that we do to the standard Transformer architecture, two of them come from the latest research work on Transformer, the third one is from our internal research group. They all, to a certain extent, help improve the baseline model, but the improvement is not consistent

* Equal contribution

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 368–376
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64035

across all languages and it becomes progressively weaker, diluted in the combination of other techniques.

### 2.1 Transformer with Relative Position Attention

We use relative position representation in self-attention mechanism (Shaw et al., 2018) of both the encoder and decoder side for all systems. Originally, the Transformer only uses the absolute position information in the word embedding layer, lacking of position information in higher layers. Incorporating explicit relative position information in self-attention enables its propagation to the higher layers. And in contrast to the absolute position, it's invariant to the sentence length. We compared the translation results between whether using this feature or not, and found that with the relative position features, the model performs better in reordering. We also implement the relative position representation with fast decoding. Experiments showed that it lead to faster convergence and better performance.

### 2.2 Weighted Transformer

**Motivation:** The Transformer Model proposed by Vaswani et al. (2017), uses a self-attention mechanism to avoid recurrence and convolution in previously proposed models. The heads in the multi-head attention are independent of each other, Ahmed et al. (2017) improved this with a new mechanism, namely multi-branch attention. The latter adds a group of dynamic learned parameters to distinguish the importance of the heads.

**Our Implementation:** We implement the weighted transformer, with extra small improvements compared to the original implementation. We introduce the weighting mechanism to both encoder and the bottom layer of multi-head attention in decoder which does not accept encoder output states. The reason we do not add in the upper layer of multi-head attention is that it causes about 3 times slower of training speed.

### 2.3 Neural Suffix Prediction for Russian

For English to Russian task, we implement Song et al. (2018) 's work, namely neural suffix prediction, in our baseline system. Song et al. (2018) 's work takes a two-step approach for the decoder. Russian word sequence is split into stem sequence and suffix sequence. During the decoding time, stem is first generated at each decoding step, before suffix is predicted. Due to limited resource, we didn't strictly evaluate the actual improvement of this method, compared with the baseline Transformer architecture. We directly use it as our baseline system. For the following part of this paper, our English to Russian model is with neural suffix prediction by default. We use Byte Pair Encoding (BPE) (Sennrich et al., 2015) to get subword sequence of English side. For Russian side, BPE is applied on the stem sequence.

## 3 System Overview

### 3.1 Large-scale Back Translation

Adding synthetic data through the process of back translation (Sennrich et al., 2016) has become the paradigm when building state-of-the-art NMT systems. especially when a large amount of target-side in-domain data is available. For low-resource languages, the use of back-translated monolingual data is crucial as the target side lexicon coverage is often insufficient, it is the case for English ↔ Turkish, with only 0.2M bilingual sentence pairs and Turkish being a very morphologically-rich language.

Considering the abundant volume of the monolingual data provided by the organizers and the costful process of back translation, we need to select among the entire monolingual data those of quality and being close to our domain of interest. We use the methods described in the data selection section (Subsection 3.2) to select this in-domain data from the large monolingual data.

Then comes the question of how many back translated data should be used. Our experiments showed that it's difficult to have an universal recipe for all languages across all tasks, we had to experimentally tune the amount of synthetic data to use according to the specific task, even for the two directions within the same language pair (See Table 1 for more details).

For different translation tasks, we use synthetic data ranging from 5 million to 70 million in combination with the provided parallel corpus to train the NMT system, resulting in an increase of +3 to +7 BLEU point over our baseline systems.

In order to understand the effectiveness of the large-scale back translation, we give a simple analysis using the example of English → Russian. A big Russian language model using 96 million monolingual data (All-96M-LM) is trained for this

| | authentic | synthetic (critical) | synthetic (upper bound tested) |
|---|---|---|---|
| **EN→ RU** | 8M | 8M | 24M |
| **RU→ EN** | 30M | 70M | 85M |
| **EN→ TR** | 0.2M | 6M | 14M |
| **TR→ EN** | 0.2M | 10M | 15M |
| **EN→ ZH** | 7.2M | 3.98M | 10M |

Table 1: Synthetic data usage. **Authentic**: the amount of authentic parallel data after cleaning; **Synthetic (critical):** the maximal amount of synthetic data added to the parallel data with improvement; **Synthetic (upper bound tested)**: the maximal amount of synthetic data tested

| | **baseline translation** | **translation with BT** | **reference translation** |
|---|---|---|---|
| | (BLEU 36.66) | (BLEU 38.94) | |
| **All-96M-LM** | 206.35 | 203.07 | 197.95 |
| **NoUN-80M-LM** | 204.72 | 199.83 | 194.83 |

Table 2: Perplexity analysis of the effect of back translation with English → Russian examples

purpose, then 3 different translations of the newstest2017 test set are evaluated in term of perplexity over this language model, the results are shown in the Table 2. We can see that the translation produced by a model using back translation has a lower perplexity than the one without using it, and an increase of +2.3 BLEU is observed accordingly. This means that with the extra target side in-domain data, the model can learn to produce more fluent translation.

Similar observations can be obtained using a different language model (NoUN-80M-LM), we can notice that without the UN data, the same translations have lower perplexity, as the UN domain is a different domain than the news one, that's also in line with the BLEU score increase when training without the UN corpora in the English → Russian experiment results (See Subsection 4.2 ).

### 3.2 Fine-tuning with In-domain Data

Fine-tuning is a common method for domain adaption in NMT, which has proven effective for boosting the translation quality in a specific domain. Following Luong and Manning (2015), we first train a model on a large out-of-domain corpus and then continue a few epochs only on a small in-domain corpus. In our work, we try two different approaches to select the small in-domain corpus, namely, n-grams and binary classification.

**N-grams:** In order to acquire high-quality in-domain data, we exploit the algorithm detailed in Duh et al. (2013); Axelrod et al. (2011), which aims at selecting sentence pairs from large out-domain corpus that are similar to the target domain. In our experiment, the parallel bi-texts and monolingual back-translation corpus are used as out-domain corpus $O$. While all available newstest sets are regarded as in-domain corpus $I$. We first train tri-gram language models over the source and target side of the in-domain corpus, respectively ($H_{I-src}$ and $H_{I-tgt}$). Then, build tri-gram language models of similar size over the random sample from the out-domain corpus ($H_{O-src}$ and $H_{O-tgt}$). Based on this, each sentence pair $s$ from $O$ is scored by the bilingual cross-entropy difference $[H_{I-src(s)} - H_{O-src(s)}] + [H_{I-tgt(s)} - H_{O-tgt(s)}]$. Finally, we sort all sentence pairs and select top n (n = 100K) sentences pair with the lowest scores to fine-tune the parameters of neural network.

**Binary Classification:** Finding the sentence pairs that are similar to the in-domain corpora can also be viewed as a text categorization problem, albeit there are only two categories here, that is, in-domain (1) and out-domain (0).

With the development of word embedding (Mikolov et al., 2013), we are now able to convert textual content into numerical representation that bears much more information than the traditional ngram-based models can, such as positional, semantic and syntactical information. In most sentences, there are parts that carry strong domain information and are very useful in determining whether a particular sentence is in-domain or out-domain, while other parts are much more general and thus less useful. To extract such key domain information from a sentence, we can use convolutional neural network (CNN) with a softmax classifier sitting in the top layer.

We follow the footstep of Chen and Huang (2016) where the Semi-Supervised CNN (SS-CNN) domain adaption method was proposed. We use our own cloud-based word2vec to train word embeddings of 300 dimensions, using all available WMT18 bilingual and monolingual corpora for the constrained translation tasks and all the corpora that we have access to for the unconstrained tasks. Similar to Chen and Huang (2016), we also make full use of conText (Johnson and Zhang, 2015) as the CNN-based text classifier, which features a stack of two independent CNNs. The inputs to the first network, which is a simple convolution layer, are bag-of-words one-hot vectors, concatenated one-hot vectors, bag-of-words word embedding vectors and concatenated word embedding vectors, respectively, which results in four output regions correspondingly. The regions are then fed to the actual CNN classifier altogether that consists of one convolution layer, one nonlinear layer, one max pooling layer as well as a softmax. Without the loss of generality, we refer the full stack as one CNN classifier. For bilingual corpora, we train two classifiers, one for each language. Each classifier is trained with pre-trained word embeddings of each sentence and the corresponding label (1 for in-domain or 0 for out-domain). During inference, the classifiers will score each new sentence pair, resulting in four scores. That is, for each language, we will have one score for the in-domain possibility and the other for out-domain. Then, we replace the entropy scores of the scoring equation used in the ngram-based approach with these four possibility scores, to work out the final score for the sentence pair.

While training the CNN classifiers, we first sample a general domain corpora with the same number of sentence pairs in the in-domain set to be used as the out-domain set. For SSCNN, an in-domain set of a few thousand sentence pairs is sufficient to find high quality in-domain sub-corpora from the general corpora. Then, we label all in-domain pairs with 1 and all sampled pairs as 0. Next, for each language, we pass all labelled sentences to a CNN classifier, where the first network scans the input with the window size of 5 and the stride size at 1 with a zero padding of 4. For the second network, we employ 500 neurons with ReLU as the activation for the nonlinear layer. The loss function we use is mean square error and the training progresses using SGD with momentum.

**Hyper-specialisation:** While the two methods described previously in this section allow us to acquire data that are close to our development set, however, only suboptimal performance is expected on the final test set, as we don't have the reference translation to perform the bilingual data selection for the final test set. Inspired by the idea of hyper-specialisation (Deng et al., 2017), we produced multiple hypotheses of the test set using our best single and ensemble models, and used them as the target side translations. By integrating the real source text and target side translation pairs of the test set as in-domain seed into the data selection process, we makes the latter aware of the test set information, thereby enables it to retrieve better in-domain bi-texts for this specific test set. Subsequently, these synthetic bi-texts can serve as train data as they are in-domain parallel data of good quality, the idea is to imitate the effect of model ensemble, but at the data level.

Finally, we replace Adam (Kingma and Ba, 2014) optimizer with SGD and use the learning rate decay, then we continue training the current best model for a few more iterations on the mixture of synthetic bi-texts and top n (n=100K) selected bilingual texts.

### 3.3 Greedy Model Selection based Ensembling

Model ensembling is a widely used technique to boost the performance of a MT system, which consists in combining the prediction of multiple models at each decode step. However, we have observed that if the single models were strong enough, very tiny improvement could be drawn from a simple combination of the top N models. Also combining brutally an increasing number of models could easily go over the resource limit even with very powerful multi-gpu machines.

In order to overcome this limit, we adopted an approach named Greedy Model Selection based Ensembling (GMSE) that we will describe in this section.

**GMSE Algorithm:** The algorithm takes as input a sorted list of N strong single models $\mathscr{L}_{\text{cand}} = \{\mathscr{M}_{0 \leq i \leq N}\}$ with N could possibly up to several hundreds, the order is typically defined by the performance on the development set. The algorithm starts with a "keep" list $\mathscr{C}_{\text{current}}$ which initially only contains the model $\mathscr{M}_0$. At each iteration, a model candidate $\mathscr{M}_{\text{i}}$, is shifted from the input $\mathscr{L}_{\text{cand}}$ and concatenated temporarily to the current "keep" list, all these models are then put through a standard model ensemble process. If the current iteration ends up with a better BLEU score, the candidate model $\mathscr{M}_{\text{i}}$ is added to the "keep" list $\mathscr{C}_{\text{current}}$. Otherwise, it is add to a "redemption" list $\mathscr{R}$. and still has a weak chance to be "redeemed" for the future iterations. One model from the "redemption" list can only be redeemed once, after which it is withdrawn definitely from the candidates. At the beginning of each iteration, a candidate model $\mathscr{M}_{\text{i}}$ could be either drawn from the beginning of the $\mathscr{L}_{\text{cand}}$ with a probability P, the end of the $\mathscr{L}_{\text{cand}}$ with a probability $\text{P}_{\text{reverse}}$, or the "redemption" list $\mathscr{R}$ with a probability $\text{P}_{\text{redeem}}$, we used [P, $\text{P}_{\text{reverse}}$, $\text{P}_{\text{redeem}}$] = [0.8, 0.1,0.1] for our experiments. The algorithm ends when the input list $\mathscr{L}_{\text{cand}}$ is empty or a certain number of stalls (10) is reached. See algorithm 1 for the pseudo-code.

**Algorithm 1** GMSE algorithm

```
Input:
    The sorted list of N single models ordered by performance on dev set: $\mathscr{L}_{cand} = \{\mathscr{M}_{0 \le i \le N}\}$;
    Number of stalls before stopping the algorithm: K;
Output:
    The best combination when stopping criterion is reached: $\mathscr{C}_{best}$;
Initialization:
    $\mathscr{C}_{current} = \{\mathscr{M}_0\}$
    $\mathscr{C}_{best} = \{\mathscr{M}_0\}$
    $\mathscr{R} = \{\}$
    $S_{best} = S_{M_0}$
    $k = 0$
while $k < K$ and $\mathscr{L}_{cand}$ is not empty: do
    if CONDITION($\mathscr{R}$, $P_{redeem}$) = True: then
        $\mathscr{M}_{cand}$ = shift $\mathscr{R}$
    else if CONDITION($P_{reverse}$) = True: then
        $\mathscr{M}_{cand}$ = pop $\mathscr{L}_{cand}$
    else
        $\mathscr{M}_{cand}$ = shift $\mathscr{L}_{cand}$
    end if
    $\mathscr{C}_{current} = \mathscr{C}_{current} \cup \{M_{cand}\}$
    $S_{current} = ensemble(C_{current})$
    if $S_{current} > S_{best}$ then
        $\mathscr{C}_{best} = \mathscr{C}_{current}$
        $S_{best} = S_{current}$
    else
        $\mathscr{C}_{current} = \mathscr{C}_{best}$
        $\mathscr{R} = \{\mathscr{M}_{cand}\} \cup R$
    end if
end while
```

As mentioned at the beginning of the section, the effect of model ensemble is diminished with strong single models, especially with fine-tuned models. In order to boost the performance, we trained independently a large number of models using different model features for transformer models as described in the Section 2 , different hyperparameters, different versions of training data and different model types, resulting in a search space which is sufficiently large and with high diversity. The greedy nature of the GMSE algorithm makes the search feasible in a relatively acceptable time limit. On the development set, this algorithm can consistently improve more +1 BLEU point over the best single model across all the language directions in which we have participated. This increase drops to only around +0.3 - +0.5 on test set.

### 3.4 Greedy Feature Selection based Reranking

We describe the greedy feature selection based reranking (GFSR) we used in WMT 2018 in this section. N-best reranking in machine translation is a common-used technology, which can improve translation quality by picking better translations from n-best list to replace the one with the highest MT model score.

**GFSR Framework:** We adopted the widely used an open-source implementation in moses (Koehn et al., 2007) of K-batched MIRA algorithm (Cherry and Foster, 2012) to rerank the nbest list. Unlike most common reranking architectures, we select the features greedily from a large feature pool, in which there are about 50+ different feature types.

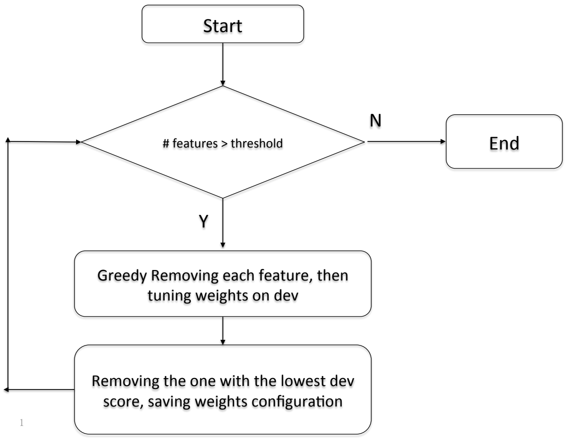

Start
# features > threshold
N
End
Y
Greedy Removing each feature, then tuning weights on dev
Removing the one with the lowest dev score, saving weights configuration
1

Figure 1: Framework of GFSR

As described in Figure 1, firstly, reranking the nbest list with all $n$ features in the feature pool. Secondly, for all features, ignoring each one of them from the feature pool in a loop, and using the other $n-1$ features to rerank on the dev data. Then, the feature that can get largest BLEU score improvement by ignoring it is removed from the

| Category | Features |
|---|---|
| NMT Model Features | Main model score |
| | Left2Right sodel score (Liu et al., 2016) |
| | Target2source model score (Sennrich et al., 2016) |
| Language Model Features | Multiple ngram language models |
| Count Features | Word count |
| | Char count |
| | Word count ratio |
| | Char count ratio |
| Word-alignment-based Features | Word posterior probability (Ueffing and Ney, 2007) |
| | Sentence-level translation probability |
| Expected Scores | Consensus score (Expected BLEU) (DeNero et al., 2009) |
| | Expected ChrF (Popović, 2015) |
| | Expected Qmean (Chen et al., 2012) |

Table 3: Features (feature templates) for reranking.

feature pools. The loop stops when the number of features is smaller than a threshold.

**Features:** We used about 50+ features in our reranking module, including NMT model features, count-based features, word-alignment-based features, expected scores features, etc. The feature types are described in Table 3. Some feature types such as NMT model features and Language Model features may have multiple instances.

**Reinforced Nbest Generation:** In order to have large beam size K = 100+ without introducing too many noises, we use multiple strong ensemble systems to generate a joint Nbest list. The idea is to have a higher upper-bound for the beam without the side-effect of having a lower lower-bound, Thereby, the reranker can focus on only good candidates.

### 3.5 Postprocessing

To recase (or recapitalize) the MT output, SMT-based recasers are trained on the Target side corpus with Moses toolkit[1]. In these models, language model plays an important role. As a result, large & domain related LMs are built. We also use a few simple uppercase rules, for example province & city names and the words beginning of a sentence are capitalized.

## 4 Experiments and Results

Preliminary experiments showed that the model features described in the section 2 yielded similar improvements reported in the original papers, or on par with the standard Transformer. For all of our baseline systems, we integrated these features into our model architecture, except the neural suffix prediction which is only used for the English → Russian system.

All of our experiments employ 6 encoder and decoder self-attention layers, both embedding and hidden size have a dimension of 512, 8 heads for the self-attention. We use FFN layer with 2048 cells and Swish (Ramachandran et al., 2017) as activation function. Warmup step is set to 16000 with a learning rate equals to 0.0003. We use label smoothing with a confidence score 0.9 and all the dropout probabilities are set to 0.1. All baseline systems are trained with 4 to 8 GPUs using synchronous-SGD with moving average mechanism where the average is taken in time and in space (Zhang et al., 2015).

We use BLEU as evaluation metric (Papineni et al., 2002). For English ↔ Russian and English ↔ Turkish, all reported scores are calculated over tokenized texts except for the 2018 submission which is end2end BLEU. For English → Chinese, all reported scores are end2end BLEU score using the SACREBLEU toolkit[2] (Post, 2018).

### 4.1 English → Chinese

For the English → Chinese system, we use all the available parallel data to train our English → Chinese system. The parallel corpus is firstly filtered using the same pipeline as for the other language pairs. As we find many sub-fragments belonging to the same translation units in the parallel data, we do an additional ngram-check based fuzzy filtering to get rid of these noisy pairs. We use an in-house tokenizer for both English and Chinese tokenization. After the preprocessing, we train BPE models with 60000 merge operations for both sides respectively.

To employ the monolingual Chinese corpus, we first build a ZH → EN Transformer system with all the available parallel data. We select the good quality in-domain corpus from the XMU monolingual corpus[3] to produce our synthetic data. The corpus contains a total number of $5,959,849$ sentences after the selection and a rule-based filtering. We set beam size as 12 and $alpha$ as 0.6 during batch-decoding. The generated synthetic data is augmented into our parallel training data to build our EN → ZH Transformer system. We extended the use of monolingual data to other sources, but it didn't result in better performance.

We follow the methods described in Subsection 3.2 for data selection. A series of models

[1] http://www.statmt.org/moses/

[2] https://github.com/awslabs/sockeye/tree/master/contrib/sacrebleu

[3] http://nlp.nju.edu.cn/cwmt-wmt/

| System | newsdev2017 | newstest2017 |
|---|---|---|
| baseline | 35.47 | 35.29 |
| + corpus cleaning | 36.02 | 36.64 |
| + back translation | 39.15 | 40.04 |
| + finetuning | 40.06 | 40.68 |
| + ensemble | 40.57 | 41.18 |
| + reranking | 40.89 | 41.60 |
| **WMT18 submission** | **43.37** | |

Table 4: EN → ZH BLEU results on *newsdev2017* and *newstest2017*

can be obtained according to the methods and the amount of data used for fine-tuning. We adopt the GMSE approach for ensemble, the final best combination contains 7 models. Our reranker contains more than 70 features, including 14 Chinese language models, 8 Target-to-Source models, 4 Right-to-Left models. We use *newsdev2017* as the development set and *newstest2017* as the validation set during model training. The results of our system are reported in Table 4.1.

### 4.2 English ↔ Russian

For English ↔ Russian, we use the following resources from the WMT parallel data: News Commentary v13, CommonCrawl, ParaCrawl corpus, Yandex Corpus, UN Parallel CorpusV1.0 and Wiki Headlines. We perform data quality assessment, language identification, and excessive BPE segmentation filtering, resulting in a 28 million high-quality bilingual data. We train bidirectional systems using this high-quality bilingual data. We use 50000 BPE operations and the vocabulary size is set to 50000. For the English → Russian system, we found that it's beneficial to not make use of the UN corpora.

We selected in-domain monolingual data using the development sets 2012-2017 as seed data from the News Crawl corpora. We back-translated 24 million Russian and 70 million English sentences into the respective source side language using the the best single model trained on the high-quality bilingual data.

### 4.3 English ↔ Turkish

All parallel training data released are used in our TR ↔ EN systems, and it is about 207K sentences. We use an in-house tokenizer for both English and Turkish tokenization. A joint BPE model is applied in both directions, which is learned from mixed corpus of EN and TR with 16000 merge operations. As the parallel data amount is small, we use a shared vocabulary for both EN and TR, and we tie all embeddings of source, target and output layer following Press and Wolf (2017).

| System | newstest2016 | newstest2017 |
|---|---|---|
| baseline | | 31.62 |
| + corpus cleaning | | 34.71 |
| + w/o UN | 31.99 | 36.15 |
| + back translation | 34.24 | 38.94 |
| + finetuning | 34.96 | 40.37 |
| + ensemble | 35.98 | 41.06 |
| + reranking | 36.41 | 41.77 |
| **WMT18 submission** | **34.8** | |

Table 5: EN → RU BLEU results on *newstest2016* and *newstest2017*

| System | newstest2016 | newstest2017 |
|---|---|---|
| baseline | 29.98 | 33.56 |
| + corpus cleaning | 30.82 | 36.33 |
| + back translation | 33.90 | 39.84 |
| + finetuning | 34.72 | 40.76 |
| + ensemble | 35.76 | 41.34 |
| + reranking | 36.23 | 41.97 |
| **WMT18 submission** | **34.9** | |

Table 6: RU → EN BLEU results on *newstest2016* and *newstest2017*

The back translation is particularly effective for EN ↔ TR as the amount of parallel data is very limited. For EN → TR, about 6 million sentences are selected from the newscrawl2016, 2017 and common crawl data, which is scored and sorted by domain similarity with newstest2016 test-set and authentic parallel data. Then, the 6 million sentences are translated into English by a TR ↔ EN model trained by the authentic parallel corpus. The domain relevance and the amount of data are important when using back-translation. The TR → EN follows the same procedure to get synthetic data, except the used monolingual data sources include news2014-2017 and news_comment, and the final amount of effective monolingual sentences is 10 million.

Unlike the back translation process, the fine-tuning is less effective as the amount of authentic parallel data is very limited. However, our data selection methods can still yield about +0.5 BLEU over strong underneath models.

| System | newstest2016 | newstest2017 |
|---|---|---|
| baseline | 14.28 | 14.97 |
| + joint-bpe | 15.83 | 16.13 |
| + corpus-cleaning | 16.31 | 16.80 |
| + back translation | 22.92 | 23.87 |
| + finetuning | 23.57 | 24.20 |
| + ensemble | 24.63 | 24.96 |
| + reranking | 25.23 | 25.76 |
| **WMT18 submission** | **20.0** | |

Table 7: EN → TR BLEU results on *newstest2016* and *newstest2017*

| System | newstest2016 | newstest2017 |
|---|---|---|
| baseline | 17.90 | 18.41 |
| + joint-bpe | 18.33 | 18.72 |
| + corpus-cleaning | 19.10 | 19.61 |
| + back translation | 26.41 | 26.98 |
| + finetuning | 27.21 | 27.52 |
| + ensemble | 28.12 | 28.04 |
| + reranking | 28.51 | 28.20 |
| **WMT18 submission** | **28.0** | |

Table 8: TR → EN BLEU results on *newstest2016* and *newstest2017*

## 5 Conclusion

This paper describes Alibaba's neural machine translation systems for the WMT18 shared news translation task. For all translation directions, we adopted the same strategies, which consist of building numerous strong single systems over which we employed reinforced multi-system based mechanisms to get the best out of all these single systems. We investigated the two mainstream methods to build a strong single system, one is based on incremental improvements of neural machine translation model architecture and the other is to have more data and make a better use of these data, and we found that the latter is more effective, at least in the cases where the former is not "revolutionary" enough. Finally, for all translation directions in which we have participated, we achieved the best results in term of case sensitive BLEU score, setting the new state-of-the-art performance.

## References

Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. 2017. Weighted transformer network for machine translation. *CoRR*, abs/1711.02132.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing*, pages 355–362. Association for Computational Linguistics.

Boxing Chen and Fei Huang. 2016. Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proceedings of The 2016 SIGNLL Conference on Computational Natural Language Learning*, pages 314–323. Association for Computational Linguistics.

Boxing Chen, Roland Kuhn, and Samuel Larkin. 2012. Port: a precision-order-recall mt evaluation metric for tuning. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 930–939. Association for Computational Linguistics.

Colin Cherry and George Foster. 2012. 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.

John DeNero, David Chiang, and Kevin Knight. 2009. Fast consensus decoding over translation forests. 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*, pages 567–575. Association for Computational Linguistics.

Yongchao Deng, Jungi Kim, Guillaume Klein, Catherine Kobus, Natalia Segal, Christophe Servan, Bo Wang, Dakun Zhang, Josep Maria Crego, and Jean Senellart. 2017. SYSTRAN purely neural MT engines for WMT2017. *CoRR*, abs/1709.03814.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 678–683.

Rie Johnson and Tong Zhang. 2015. Semi-supervised convolutional neural networks for text categorization via region embedding. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 919–927. Curran Associates, Inc.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Matt Post. 2018. A call for clarity in reporting BLEU scores. *CoRR*, abs/1804.08771.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *EACL*.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2017. Searching for activation functions. *CoRR*, abs/1710.05941.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *CoRR*, abs/1803.02155.

Kai Song, Yue Zhang, Min Zhang, and Weihua Luo. 2018. Improved english to russian translation by neural suffix prediction. *CoRR*, abs/1801.03615.

Nicola Ueffing and Hermann Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Sixin Zhang, Anna E Choromanska, and Yann LeCun. 2015. Deep learning with elastic averaging sgd. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 685–693. Curran Associates, Inc.

# The RWTH Aachen University English-German and German-English Unsupervised Neural Machine Translation Systems for WMT 2018

**Miguel Graça, Yunsu Kim, Julian Schamper, Jiahui Geng and Hermann Ney**
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
{graca,kim,schamper,jgeng,ney}@i6.informatik.rwth-aachen.de

## Abstract

This paper describes the unsupervised neural machine translation (NMT) systems of the RWTH Aachen University developed for the English ↔ German news translation task of the *EMNLP 2018 Third Conference on Machine Translation* (WMT 2018). Our work is based on iterative back-translation using a shared encoder-decoder NMT model. We extensively compare different vocabulary types, word embedding initialization schemes and optimization methods for our model. We also investigate gating and weight normalization for the word embedding layer.

## 1 Introduction

Unsupervised NMT was recently investigated in (Artetxe et al., 2017; Lample et al., 2017, 2018) and has shown promising results in language pairs like German to English. For the WMT 2018 unsupervised learning track, we combine the concepts proposed in previous research and perform a thorough comparison of the main components of each method. Additionally, we augment the word embedding initialization with weight normalization to improve its integration in the model and with a gating technique to allow the model to learn task specific information.

The main findings of this paper are: (i) the iterative method (Lample et al., 2017) outperforms the online training method (Artetxe et al., 2017), (ii) cross-lingual embedding initialization is required in the online method and (iii) byte-pair encoding (BPE)-based vocabularies (Sennrich et al., 2016) outperform word-based vocabularies in online training.

This paper is organized as follows: Section 2 describes pre- and postprocessing pipelines, corpora selection and vocabularies used in our experiments. Section 3 details the models used in this work together with the embedding augmentation techniques. The experimental evaluation is presented in Sections 4 and 5 and finally we conclude with Section 6.

## 2 Pre- and Postprocessing

Our preprocessing pipeline consists of a tokenization with a script from the Moses toolkit (Koehn et al., 2007), lower-casing, and the introduction of a number category token which replaces all occurrences of numbers. We use joint BPE in our experiments and apply it at this stage of preprocessing.

After the search procedure, we first monotonically replace number tokens with their original content, and unknown words to the target hypothesis by their order of occurrence in the source sentence. This method is very restrictive, as it fails when, e.g., more unknown tokens are hypothesized than there are in the source sentence due to an unknown token being attended twice. Since, to our knowledge, there are no well-founded methods of pin-pointing which source words are attended during the generation of a target word in the Transformer (Vaswani et al., 2017), we decided for the forementioned method.

As postprocessing, we first convert subwords to words. Lower-cased words are then frequent-cased using the tools provided in the Jane toolkit (Vilar et al., 2010). As a final step, the text is detokenized using the detokenizer from Moses and punctuation is normalized.

### 2.1 Corpora Selection

We use monolingual News Crawl articles from 2014 to 2017[1] as our training corpora for both German and English languages. $100M$ sentences are sub-sampled for pre-training word embeddings and $5M$ sentences are used for translation model training.

[1] http://www.statmt.org/wmt18/translation-task.html

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 377–385
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64036

|  | German | English |
|---|---|---|
| # sentences | 5M | 5M |
| Vocabulary | 1.3M | 577K |
| OOV rate | 6.9% / 8.9% | 1.73% / 3.3% |
| Effective voc. | 50K / 46.3K | 50K / 31.2K |

Table 1: Corpus statistics for the German and English monolingual corpora. OOV word rates and effective vocabulary sizes are given for unshared and shared, respectively displayed, vocabularies limited to the most frequent $50k$ words.

Table 1 shows the corpus statistics for the model training. The reported out-of-vocabulary (OOV) word rates and effective vocabulary sizes are shown for our word-level experiments, which use either top frequent $50k$ words for each language or a shared vocabulary with a total of $50k$ words.

Even though the News Crawl corpora contain mostly clean data, we noticed that common English words are found in the German corpus and vice-versa, which causes an overlap in the effective vocabulary.

### 2.2 Vocabulary

In this work we consider different kinds of vocabularies for the unsupervised translation systems. Lample et al. (2017) use word-level vocabularies due to the initialization with a word-by-word model. Artetxe et al. (2017) perform experiments on both word- and BPE-level and report that the model has difficulties translating rare sub-word codes. We perform experiments with both BPE and word vocabularies to find the best setting for unsupervised NMT.

For the BPE vocabularies, we consider only the joint variant, performing $20k$ and $50k$ merge operations. The word-level vocabulary is restricted to a $50k$-word shared vocabulary or two seperate $50k$-word German and English vocabularies.

## 3 Unsupervised Neural Machine Translation

The models proposed in Lample et al. (2017); Artetxe et al. (2017) follow a recurrent attention-based encoder-decoder architecture (Bahdanau et al., 2015). As a follow-up work Lample et al. (2018) make use of the Transformer (Vaswani et al., 2017) architecture, which we also utilize for our systems.

### 3.1 Model Description

We closely follow the model architecture in (Lample et al., 2017), but with a Transformer encoder-decoder. It is able to translate in both source to target and target to source translation directions via joint training and parameter sharing of components, therefore we denote it further as a shared architecture. In this section, we describe how the model functions for an input source sentence $f_1^J = f_1, ..., f_j, ..., f_J$ and output target sentence $e_1^I = e_1, ..., e_i, ..., e_I$.

The model consists of an self-attentive encoder and decoder, word embeddings and output layers, where the encoder and decoder share parameters in both translation directions. The output layer may additionally be shared when the output vocabularies are also shared between both directions.

**Word embeddings**: Each word is encoded in a continuous space of dimension $D$ via a lookup table function $E : V \to \mathbb{R}^D$, where $V$ represents the source or target vocabulary, scaled up by $\sqrt{D}$ as in the original formulation (Vaswani et al., 2017). Fixed positional embeddings $pos : \mathbb{N}_0 \to \mathbb{R}^D$ (Vaswani et al., 2017), which encode the absolute position $j$ of a word $f_j$ in the source sentence, are added to the word vectors to represent a word embedding:

$$\bar{f}_j = E_f(f_j) \cdot \sqrt{D} + pos(j) \qquad (1)$$

Source word embeddings are applied whenever the model reads a source sentence or outputs a source sentence. All of the above hold analogously for the target word embeddings.

**Encoder**: The input source embeddings are read by a self-attentive encoder module and outputs a sequence of hidden states $h_1^J$ with $h_j \in \mathbb{R}^D$ having the same dimensionality as the input embeddings.

$$h_1^J = H(\bar{f}\,_1^J; \theta_{enc}) \qquad (2)$$

A noise model as described in (Lample et al., 2017) is applied to the encoder inputs.

**Decoder**: Target word predictions are conditioned on the sequence of previously seen embedded target words $\bar{e}\,_0^{i-1}$ and the encoder outputs $h_1^J$. The decoder outputs a single hidden state $s_i \in \mathbb{R}^D$, which is then propagated to an output layer. Note that in our setup encoder and decoder outputs have the same dimensionality.

$$s_i = S(h_1^J, \bar{e}\,_0^{i-1}; \theta_{dec}) \qquad (3)$$

The target sentence is augmented with a sentence start symbol $e_0$, which is an identifier for the output language. In our setup the decoder is shared between languages.

**Output layer**: The hidden state $s_i$ is projected to the size of the output vocabulary and normalized with a softmax operation resulting in a probability distribution over target words.

$$p(e_i|e_1^{i-1}, f_1^J) = \text{softmax}(W \cdot s_i + b)_{e_i} \tag{4}$$

As mentioned in Section 2.2, the output layer may or may not be shared depending on the type of vocabularies.

**Optimization**: The model is trained via cross-entropy on both translation directions. Additionally, we include auto-encoding losses for both languages for a total of four optimization criteria as in both approaches (Artetxe et al., 2017; Lample et al., 2017).

Finally, we include an adversarial loss term (Lample et al., 2017) in a feature study experiment, where the model is trained to fool a separate model that attempts to discriminate the language of the input sentence after the encoder module. Each component of the loss function is equally weighted.

Note that, in contrast to (Artetxe et al., 2017), we do not alternate between loss functions during optimization and instead optimize the summation of them. We noticed the same translation quality when comparing both sum and alternating variants in preliminary experiments.

### 3.2 Batch Optimization

Proposed by Lample et al. (2017), the batch optimization method trains the model iteratively: the model trained on iteration $n-1$ is used to generate back-translations to train the model at iteration $n$. The initial model is an unsupervised word-by-word translation model based on cross-lingual word vectors (Conneau et al., 2017).

The workflow of this method for the $n$-th iteration is as follows:

1. Translate monolingual corpora with the model at iteration $n-1$
2. Train for one epoch on the back-translated and monolingual corpora

Throughout this work, we denote an iteration as the forementioned steps. We restrict ourselves to only one epoch for model training per iteration, but it is also possible to train for a different amount of updates.

### 3.3 Online Optimization

Leveraging the model's ability to translate in both translation directions, Artetxe et al. (2017); Lample et al. (2018) generate back-translations for each mini-batch using the currently trained parameters. This method is not initialized with a word-by-word translation.

We noticed that with the original implementation training was slow due to the generation of back-translations with a smaller batch size than what fit in our device's memory. Therefore, we implement online optimization by generating 10 mini-batches of back-translations at once. We noticed no loss of translation quality when doing this.

### 3.4 Gated Word Embeddings

The initialization of the word embeddings with pre-trained word vectors allows the model to start from a much more informative state and exploit information from a larger corpus. Indeed it is a crucial component of the shared architecture, as shown empirically in Section 5.2. As an alternative to just training the initialized vector, we consider a gating mechanism, shown in Equation 5 and introduced in (Yang et al., 2016):

$$\begin{aligned}\bar{f}_j = \Big( \quad & g(f_j) \odot E_{f,pre-train}(f_j) \\ & + (1 - g(f_j)) \odot E_{f,random}(f_j) \Big) \\ & \cdot \sqrt{D} + pos(j)\end{aligned} \tag{5}$$

with the interpolation weights $g(f_j) \in \mathbb{R}^D$ being defined as a feed-forward projection to the word embeddings' dimensionality with a sigmoidal output:

$$\begin{aligned}g(f_j) = \sigma(b + W \cdot \Big[ & E_{f,pre-train}(f_j), \\ & E_{f,random}(f_j)\Big])\end{aligned} \tag{6}$$

$\odot$ denotes element-wise multiplication. This allows the model to learn task-specific information and interpolate it with the pre-trained parameters. When using this approach, the pre-trained vectors are not updated during training.

Ding and Duh (2018) perform a simpler approach to combine both kinds of embeddings, in which they concatenate the word vectors and, as in this work, keep the pre-trained embeddings fixed during training.

Our idea is most similar to the concept in (Yang et al., 2018), where the authors also employ a gating mechanism on the embeddings, but combine it with the output of the encoder in order to reinforce a language-independent encoder representation.

### 3.5 Embedding Weight Normalization

The training criteria for word embeddings does not enforce normalization constraints on the continuous output values and therefore might cause very high or low gradient values in the encoder and decoder parameters, especially at the beginning of training.

Weight normalization (Salimans and Kingma, 2016), as shown in Equation 7, normalizes each word embedding by its $L_2$-norm and introduces an additional tunable parameter $v_{f_j}$ for each word, that rescales the vector. It is initialized with the value of 1.

$$\bar{f}_j = \frac{v_{f_j} \cdot E_f(f_j) \cdot \sqrt{D}}{||E_f(f_j)||_2} + pos(j) \qquad (7)$$

## 4 Experimental Setup

All processing steps and experiments were organized with Sisyphus (Peter et al., 2018) [2] as workflow manager.

### 4.1 Model Hyperparameters

Our models use the Transformer architecture (Vaswani et al., 2017) implemented in Sockeye (Hieber et al., 2017), based on MXNet (Chen et al., 2015). The encoder and decoder both have 4 layers of size 300 with the internal feed-forward operation having 2048 nodes. The multi-head attention mechanism uses 6 heads. For each encoder and decoder layer, 10% dropout (Srivastava et al., 2014) and layer normalization (Ba et al., 2016) are used as preprocessing[3] operations and a residual connection (He et al., 2016) is additionally included in the postprocessing operations.

Monolingual word embeddings have a dimensionality of 300 and are trained as a skip-gram model using FastText (Bojanowski et al., 2017), only for words that have occured at least 10 times. Cross-lingual word embeddings are trained with MUSE (Conneau et al., 2017) for 10 epochs with the adversarial setting and 10 steps of the refinement procedure using the learned monolingual embeddings.

Model optimization is performed with the AdaM (Kingma and Ba, 2014) algorithm using a learning rate of $10^{-4}$ and a momentum parameter $\beta_1 = 0.5$. Training sequences are limited to 50 words or subwords. Parameters are initialized with Glorot initialization (Glorot and Bengio, 2010). The batch method is trained for 5 iterations, $800K$ updates, for a total of 6 days and the online method is trained for roughly the same amount of time for $500K$ updates.

Translation is performed using beam search with beam size 5 and the best hypothesis is the one with the lowest length normalized negative log-probability. Length normalization divides the sentence score by the number of words.

### 4.2 Evaluation

We constrain our results to the newstest2017 and newstest2018 data sets in the German → English translation direction. BLEU (Papineni et al., 2002), computed with mteval from the Moses toolkit (Koehn et al., 2007), and TER (Snover et al., 2006), computed with TERCom, are used as evaluation metrics. BLEU scores are case-sensitive and TER is scored lower-cased. All presented scores are percentages. For the experiments in Sections 5.3 and 5.4 we additionally test for statistical significance with MultEval (Clark et al., 2011).

Lample et al. (2017) propose a model selection criterion based on round-trip BLEU scores, however we do not notice a correlation of this measure and BLEU between experiments. The more expressive the model is, the better round-trip BLEU scores it will get, whereas BLEU itself does not change. Therefore we choose to validate on newstest2015 on the German → English translation direction for the feature study.

For our final submission, we select optimization method, embedding initialization and vocabulary types based on BLEU on the German → English direction of newstest2017 and select the best hyperparameter settings using the metric from Lample et al. (2017). In this case, we only consider models that have trained exactly 6 iterations.

## 5 Experimental Results

### 5.1 Translation Units

We experiment with both words and BPE subwords as initial work (Artetxe et al., 2017; Lample

[2] `https://github.com/rwth-i6/sisyphus`

[3] Pre- and postprocessing terminology is described in (Hieber et al., 2017).

|  |  | newstest2017 |  | newstest2018 |  |
|---|---|---|---|---|---|
|  | method | BLEU | TER | BLEU | TER |
| words | batch | **14.9** | **72.7** | **18.1** | **67.1** |
| unshared |  | 14.5 | 73.3 | 17.2 | 67.8 |
| words | online | 11.9 | 75.7 | 14.2 | 71.0 |
| unshared |  | 10.6 | 77.7 | 13.2 | 73.1 |
| BPE 20k |  | 11.8 | 77.9 | 13.6 | 73.9 |
| BPE 50k |  | **13.1** | **75.5** | **15.4** | **70.8** |

Table 2: Vocabulary comparison between different optimization methods for German → English. All systems are initialized with cross-lingual word embeddings.

et al., 2017) focuses primarily on words and only briefly discuss the effects of sub-word units.

Table 2 shows the effect of different vocabulary sizes and units on both online and batch optimization methods. The best performing experiments are trained with batch optimization and a word-based vocabulary, even though they face an OOV word problem during both training and testing. Furthermore restricting the vocabulary to the top-$50k$ most common words in both vocabularies and sharing an output layer performs up to $0.9\%$ BLEU and $0.7\%$ TER better than using separate top-$50k$ vocabularies (different output layers). The same effect is noticeable with the online optimization method.

The online optimization method is additionally run with joint BPE codes trained with $20k$ and $50k$ merge operations, which improves over the word-based vocabulary by up to $1.2\%$ BLEU and $0.2\%$ TER when using $50k$ operations.

We do not present results with the batch method and BPE-based vocabularies, because the initial word-by-word translation is designed to work on the word-level.

### 5.2 Embedding Initialization

Initializing word embeddings with pre-trained vectors was a component in both original works (Artetxe et al., 2017; Lample et al., 2017). Two kinds of embeddings are considered, monolingual and cross-lingual, both serving the role of initializing the model with prior knowledge to aid the training of the model. Cross-lingual embeddings further add the property of language abstraction to pre-trained monolingual vectors.

Results on the embedding initialization are reported in Table 3 for both batch and online optimization methods.

|  |  | newstest2017 |  | newstest2018 |  |
|---|---|---|---|---|---|
|  | method | BLEU | TER | BLEU | TER |
| random | online | 4.9 | 92.7 | 4.9 | 91.7 |
| monolingual |  | 7.5 | 88.2 | 8.2 | 85.7 |
| cross-lingual |  | **13.1** | **75.5** | **15.4** | **70.8** |
| + frozen |  | 12.7 | 76.3 | 15.1 | 71.6 |
| random | batch | 14.5 | 73.6 | 17.6 | 68.2 |
| monolingual |  | 14.3 | 73.3 | 17.2 | 68.0 |
| cross-lingual |  | **14.9** | **72.7** | **18.1** | **67.1** |
| + frozen |  | 14.0 | 75.8 | 16.9 | 71.5 |

Table 3: Embedding initialization comparison between different optimization methods for German → English. Online systems use joint BPE with $50k$ merge operations, whereas batch systems use seperate word-based vocabularies. Word-by-word initialization is only used for the batch optimized system.

First considering the online optimization scenario, both random and monolingual initializations fail to produce proper results. This is due to the differing word distributions for source and target embeddings that are given as an input to the encoder and decoder modules. Once the embeddings are language-independent, the model is able to achieve much better values. This follows the same motivation as the adversarial feature proposed by Lample et al. (2017), where the authors argue that the decoder must be fed with language-independant inputs in order to function effectively. Freezing the embeddings during training is also detrimental to translation quality.

Examining the initialization with the batch optimization method results in similar behaviours for a cross-lingual initialization. Here the initialization has a slight, albeit significant, influence on the translation quality. This is due to the cross-lingual signal already being strongly present in the word-by-word initialization, replacing the prior information that one gets from the word embedding initialization. Random and monolingual initializations perform roughly the same, which shows again the problem with the differing representation distributions. Overall, the cross-lingual initialization performs best for both methods.

Recently, Lample et al. (2018) have noted that it is possible to share embeddings across languages and initialize them with monolingual word vectors. We leave this for future work.

### 5.3 Embedding Features

Considering the empirical results of the previous section, we focus on improving upon the integra-

|  | newstest2017 |  | newstest2018 |  |
|---|---|---|---|---|
|  | BLEU | TER | BLEU | TER |
| baseline | **14.9** | 72.7 | 18.1 | **67.1** |
| + frozen emb. | 14.0* | 75.8* | 16.9* | 71.5* |
| + gating | 14.4* | **72.5** | 17.6* | 67.3 |
| + emb. WN | 14.5* | 73.4* | 17.5* | 68.4* |
| + emb. WN | 14.7 | 72.8 | 18.2 | **67.1** |

Table 4: Results for different embedding initialization on systems optimized with the online strategy for German → English. The baseline system uses batch optimization, cross-lingual embeddings and shared vocabularies. WN stands for weight normalization. * denotes a p-value of $< 0.01$ w.r.t. the baseline.

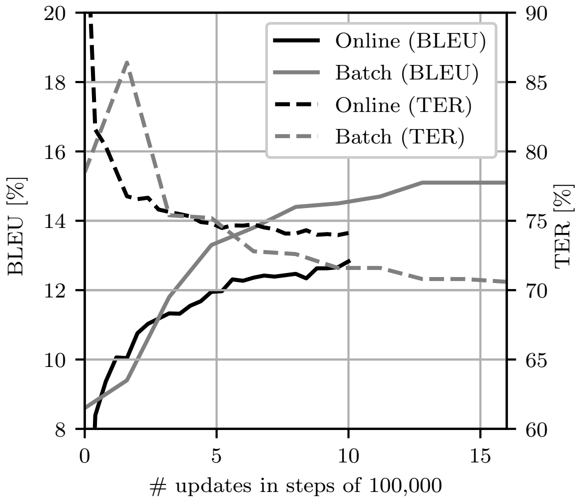


Figure 1: BLEU and TER values on newstest2017 German → English for checkpoint models of online and batch optimization methods. The initial step of the batch method uses the word-by-word translation scores.

tion of pre-trained embeddings on top of the best system, namely a word-based batch optimized system with cross-lingual embeddings and a shared output layer. Experiments are presented in Table 4.

As seen in Table 3, freezing embeddings during training worsens translation quality. One can conclude that the model learns task-specific information via the embedding component.

As a first step, we apply the gating mechanism as presented in Section 3.4 and observe an increase in performance of up to 0.7% BLEU . However, results show that the model performs up to 0.5% BLEU worse than the baseline and achieves roughly the same TER performance.

Afterwards, we apply weight normalization as presented in Section 3.5 on top of both trainable and frozen embeddings. When applied on top of frozen embeddings, the normalization helps, but still lags behind the baseline. Adding it on top of the baseline does not worsen, but also does not improve translation quality.

These experiments allow one to conclude that fine-tuning embeddings to the task at hand performs better than the implemented techniques. Alternative embedding features could be considered, as for example the works mentioned in Section 3.4.

### 5.4 Training Variations

In this Section, we consider additional experiments that do not fit in a specific category and present them in Table 5.

Firstly, we add an adversarial loss term as in (Lample et al., 2017) on top of a batch optimized model with cross-lingual embeddings and a shared output layer. We report that performance drops by up to 1.2% BLEU and we hypothesize that the feature does not integrate well in the Transformer architecture. Specifically, the encoder outputs of an LSTM (Hochreiter and Schmidhuber, 1997) are bounded between -1 and 1, whereas the Transformer encoder outputs can take on any real value. The effect of the feature was not reproducible in separate experiments with the setup described in the original publication.

Secondly, we separate both output layer and decoder components from the model to obtain a setting similar to the one in (Artetxe et al., 2017). Translation quality drops by up to 0.8% BLEU and 0.9% TER . Note that in Section 5.1, we already saw a drop of roughly the same amount when not sharing the output layer.

We investigate whether noisy input sentences and auto-encoding are necessary at later stages of the training. Hence, these features are disabled after the 3rd iteration. The improvements are not statistical significant, but at the very least the comparison shows that the model does not worsen from focusing solely on the translation task after its initial learning period. This is due to it already being able to generate decent translations after the first few iterations.

Finally, we train the batch and online methods for a larger number of iterations, see Table 6, reaching 19.2% BLEU with the batch method af-

|  | newstest2017 |  | newstest2018 |  |
|---|---|---|---|---|
|  | BLEU | TER | BLEU | TER |
| baseline | 14.9 | 72.7 | 18.1 | 67.1 |
| + adversarial | 13.9* | 74.2* | 16.9* | 69.0* |
| + unshared decoder | 14.3* | 73.3* | 17.3* | 68.0* |
| + drop AE & noise | **15.2** | **72.6** | **18.3** | **66.9** |

Table 5: Results for training variations on German → English. The baseline system uses batch optimization, cross-lingual embeddings and shared vocabularies. * denotes a p-value of $< 0.01$ w.r.t. the baseline.

|  | newstest2018 |  |  |  |
|---|---|---|---|---|
|  | De → En |  | En → De |  |
|  | BLEU | TER | BLEU | TER |
| online method | 15.4 | 70.8 | 12.0 | 79.5 |
| $1M$ updates | **16.8** | **69.3** | **13.2** | **77.7** |
| batch method | 18.1 | 67.1 | 14.0 | 77.0 |
| 10th iteration | **19.2** | **64.6** | **15.4** | **74.3** |

Table 6: Results for longer training iterations for German ↔ English. The baseline system uses batch optimization, cross-lingual embeddings and shared vocabularies.

ter 10 iterations and $16.8\%$ BLEU with the online method after $1M$ updates on newstest2017. The extended training for the online and batch method trained for 14 and 12 days respectively. Figure 1 shows the training of the models on newstest2017 German → English. The initial TER spike occurs because hypotheses are $13\%$ longer than the ones of the word-by-word system. Considering the results of these experiments, one should look into better optimization algorithm tuning for the online method.

### 5.5 Final Submission

The model in the final submission, shown in Table 7, consists of a word-based model with separate vocabularies, trained with the batch optimization method, initialized with cross-lingual embeddings, applies embedding weight normalization and is trained with a learning rate of $3 \cdot 10^{-4}$. The ensemble system consists of 4 variations of the single-best model, varying in learning rate values ($3 \cdot 10^{-4} \rightarrow 10^{-4}$), feed-forward projection hidden sizes (2048 → 1024) and monolingual, instead of cross-lingual, embedding initialization.

For reference, we include our supervised submission system for the German → English constrained task. As expected, there is a large performance gap between both our systems. The most crucial point of improvement in our submission is the amount of data used. We used a small amount of the available data, even smaller than for our supervised submission, since we noted that the models took a long time to converge as portrayed in Figure 1. We suggest to invest efforts into tuning of optimization algorithm hyperparameters and using more data.

## 6 Conclusion

The RWTH Aachen University has participated in the WMT 2018 German → English and English → German unsupervised news translation tasks. We focus on reproducing related work and infer empirically that the batch optimization method from (Lample et al., 2017) performs best on our constrained setting, i.e. $5M$ sentences for each language. An initialization with cross-lingual word embeddings performs best for both optimization strategies. Sharing vocabularies is important for a shared model architecture. BPE-based vocabularies outperform word-based ones with the online optimization method. The noise model and auto-encoding losses are not needed in later stages of training in batch optimization. Freezing the word embedding layer during training hurts. Simply initializing and training the embeddings performs better than performing weight normalization or applying a gating mechanism on top of a frozen embedding layer.

## Acknowledgements

This work has received funding from the European Research Council (ERC) (under the European Union's Horizon 2020 research and innovation programme, grant agreement No 694537, project ”SEQCLAS”) and the Deutsche Forschungsgemeinschaft (DFG; grant agreement NE 572/8-1, project ”CoreTec”). The GPU computing cluster was supported by DFG (Deutsche Forschungsgemeinschaft) under grant INST 222/1168-1 FUGG.

The work reflects only the authors' views and none of the funding agencies is responsible for any use that may be made of the information it contains.

| | German → English | | | | | | English → German | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **newstest2016** | | **newstest2017** | | **newstest2018** | | **newstest2016** | | **newstest2017** | | **newstest2018** | |
| | Bleu | Ter | Bleu | Ter | Bleu | Ter | Bleu | Ter | Bleu | Ter | Bleu | Ter |
| Single-best | 17.2 | 68.7 | 14.5 | 72.9 | 18.1 | 66.9 | 13.7 | 77.0 | 11.2 | 82.0 | 14.5 | 75.8 |
| Ensemble of 4 | 17.6 | 68.3 | 14.9 | 72.1 | 18.5 | 67.0 | 14.1 | 76.4 | 11.5 | 81.6 | 15.0 | 74.7 |
| WMT 2018 Supervised submission | 46.0 | 41.0 | 39.9 | 47.6 | 48.4 | 38.1 | - | - | - | - | - | - |

Table 7: Submission systems for the WMT 2018 German ↔ English news translation task.

## References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*. Version 2.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*. Version 1.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations*, pages 1–14, San Diego, California, USA.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*. Version 1.

Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. 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, pages 176–181.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*. Version 3.

Shuoyang Ding and Kevin Duh. 2018. How do source-side monolingual word embeddings impact neural machine translation? *arXiv preprint arXiv:1806.01515*. Version 2.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1712.05690*. Version 2.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. Version 9.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043. Version 2.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*. Version 1.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Jan-Thorsten Peter, Eugen Beck, and Hermann Ney. 2018. Sisyphus, a workflow manager designed for machine translation and automatic speech recognition. In *Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

Tim Salimans and Diederik P Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, volume 200.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open source hierarchical translation, extended with reordering and lexicon models. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 262–270.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Unsupervised neural machine translation with weight sharing. *arXiv preprint arXiv:1804.09057*. Version 1.

Zhilin Yang, Bhuwan Dhingra, Ye Yuan, Junjie Hu, William W Cohen, and Ruslan Salakhutdinov. 2016. Words or characters? fine-grained gating for reading comprehension. *arXiv preprint arXiv:1611.01724*. Version 2.

# Cognate-aware morphological segmentation for multilingual neural translation

**Stig-Arne Grönroos**
stig-arne.gronroos@aalto.fi
Aalto University, Finland

**Sami Virpioja**
sami.virpioja@aalto.fi
Aalto University, Finland
Utopia Analytics, Finland

**Mikko Kurimo**
mikko.kurimo@aalto.fi
Aalto University, Finland

## Abstract

This article describes the Aalto University entry to the WMT18 News Translation Shared Task. We participate in the multilingual subtrack with a system trained under the constrained condition to translate from English to both Finnish and Estonian. The system is based on the Transformer model. We focus on improving the consistency of morphological segmentation for words that are similar orthographically, semantically, and distributionally; such words include etymological cognates, loan words, and proper names. For this, we introduce Cognate Morfessor, a multilingual variant of the Morfessor method. We show that our approach improves the translation quality particularly for Estonian, which has less resources for training the translation model.

## 1 Introduction

Cognates are words in different languages, which due to a shared etymological origin are represented as identical or nearly identical strings, and also refer to the same or similar concepts. Ideally the cognate pair is similar orthographically, semantically, and distributionally. Care must be taken with "false friends", i.e. words with similar string representation but different semantics. Following usage in Natural Language Processing, e.g. (Kondrak, 2001), we use this broader definition of the term cognate, without placing the same weight on etymological origin as in historical linguistics. Therefore we accept loan words as cognates.

In any language pair written in the same alphabet, cognates can be found among names of persons, locations and other proper names. Cognates are more frequent in related languages, such as Finnish and Estonian. These additional cognates are words of any part-of-speech, which happen to have a shared origin.

In this work we set out to improve morphological segmentation for multilingual translation systems with one source language and two related target languages. One of the target languages is assumed to be a low-resource language. The motivation for using such a system is to exploit the large resources of a related language in order to improve the quality of translation into the low-resource language.

Consistency of the segmentations is important when using subword units in machine translation. We identify three types of consistency in the multilingual translation setting (see examples in Table 1):

(i) The benefit of consistency is most evident when the translated word is an identical cognate between the source and a target language. If the source and target segmentations are consistent, such words can be translated by sequentially copying subwords from source to target.

(ii) Language-internal consistency means that when a subword boundary is added, its location corresponds to a true morpheme boundary, and that if some morpheme boundaries are left unsegmented, the choices are consistent between words. This improves the productivity of the subwords and reduces the risk of introducing short, word-internal errors at the subword boundaries. In the example **saami + miseksi*, choosing the wrong second morph causes the letters *mi* to be accidentally repeated.

(iii) When training a multilingual model, a third form of consistency arises between the different target languages. An optimal segmentation would maximize the use of morphemes with cross-lingually similar string rep-

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 386–393
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64037

| type | consistent | en | fi | et |
|---|---|---|---|---|
| (i) | yes | On + y + sz + kie + wicz | On + y + sz + kie + wicz | On + y + sz + kie + wicz |
| (ii) | yes | gett + ing | saa + mise + ksi | saa + mise + ks |
| | | work + ing | toimi + mise + ksi | toimi + mise + ks |
| (iii) | yes | work time | työ + aja + sta | töö + aja + st |
| (i) | no | On + y + sz + kie + wicz | Onys + zk + ie + wi + cz | O + nysz + ki + ewicz |
| (ii) | no | get + ting | saami + seksi | saami + seks |
| | | work + ing | toimi + mise + ksi | toimi + miseks |
| (iii) | no | work time | työ + aja + sta | tööajast |

Table 1: Example consistent and inconsistent segmentations.

resentations and meanings, whether they occur in cognate words or elsewhere. We hypothesize that segmentation consistency between target languages enables learning of better generalizing subword representations. This consistency allows contexts seen in the high-resource corpus to fill in for those missing from the low-resource corpus. This should lead to improved translation results, especially for the lower resourced target language.

Naïve joint training of a segmentation model, e.g. by training Byte Pair Encoding (BPE) (Sennrich et al., 2015) on the concatenation of the training corpora in different languages, can only address consistency when the cognates are identical (type *i*), or with some luck if the differences occur in the ends of the words. If a single letter changes in the middle of a cognate, consistent subwords that span over the location of the change are found only by chance. In order to encourage stronger consistency, we propose a segmentation model that uses automatically extracted cognates and fuzzy matching between cognate morphs.

In this work we also contribute two new features to the OpenNMT translation system: Ensemble decoding, and fine-tuning a pre-trained model using a compatible data set.[1]

### 1.1 Related work

Improving segmentation through multilingual learning has been studied before. Snyder and Barzilay (2008) propose an unsupervised, Bayesian method, which only uses parallel phrases as training data. Wicentowski (2004) present a supervised method, which requires lemmatization. The method of Naradowsky and Toutanova (2011) is also unsupervised, utilizing a hidden semi-Markov model, but it requires rich features on the input data.

The subtask of cognate extraction has seen much research effort (Mitkov et al., 2007; Bloodgood and Strauss, 2017; Ciobanu and Dinu, 2014). Most methods are supervised, and/or require rich features.

There is also work on cognate identification from historical linguistics perspective (Rama, 2016; Kondrak, 2009), where the aim is to classify which cognate candidates truly share an etymological origin.

We propose a language-agnostic, unsupervised method, which doesn't require annotations, lemmatizers, analyzers or parsers. Our method can exploit both monolingual and parallel data, and can use cognates of any part-of-speech.

## 2 Cognate Morfessor

We introduce a new variant of Morfessor for cross-lingual segmentation.[2] It is trained using a bilingual corpus, so that both target languages are trained simultaneously.

We allow each language to have its own subword lexicon. In essence, as a Morfessor model consists of a lexicon and the corpus encoded with that lexicon, we now have two separate complete Morfessor sub-models. The two models are linked through the training algorithm. We want the segmentation of non-cognates to tend towards the normal Morfessor Baseline segmentation, but place some additional constraints on how the cognates are segmented.

In our first experiments, we only restricted the number of subwords on both sides of the cognate pair to be equal. This criterion was

[1] Our changes are awaiting inclusion in OpenNMT. In the mean time, they are available from `https://github.com/Waino/OpenNMT-py/tree/ensemble`

[2] Available from `https://github.com/Waino/morfessor-cognates`

too loose, and we saw many of the longer cognates segmented with both 1-to-N and N-to-1 morpheme correspondences. For example

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| ty | + | ö | + | aja | + | sta |
| töö | + | aja | + | s | + | t |

To further encourage consistency, we included a third component to the model, which encodes the letter edits transforming the subwords of one cognate into the other.

Cognate Morfessor is inspired by Allomorfessor (Kohonen et al., 2009; Virpioja et al., 2010), which is a variant of Morfessor that includes modeling of allomorphic variation. Simultaneously to learning the segmentations, Allomorfessor learns a lexicon of transformations to convert a morph into one of its allomorphs. Allomorfessor is trained on monolingual data.

We implement the new version as an extension of Morfessor Baseline 2.0 (Virpioja et al., 2013).

### 2.1 Model

The Morfessor Baseline cost function (Creutz and Lagus, 2002)

$$\mathrm{L}(\boldsymbol{\theta}, \boldsymbol{D}) = -\log p(\boldsymbol{\theta}) - \log p(\boldsymbol{D} \mid \boldsymbol{\theta}) \qquad (1)$$

is extended to

$$\begin{aligned}\mathrm{L}(\boldsymbol{\theta}, \boldsymbol{D}) = & -\log p(\boldsymbol{\theta}_1) - \log p(\boldsymbol{\theta}_2) - \log p(\boldsymbol{\theta}_E) \\ & - \log p(\boldsymbol{D}_1 \mid \boldsymbol{\theta}_1) - \log p(\boldsymbol{D}_2 \mid \boldsymbol{\theta}_2) \\ & - \log p(\boldsymbol{D}_E \mid \boldsymbol{\theta}_E) \end{aligned} \qquad (2)$$

dividing both lexicon and corpus coding costs into three parts: one for each language ($\boldsymbol{\theta}_1, \boldsymbol{D}_1$ and $\boldsymbol{\theta}_2, \boldsymbol{D}_2$) and one for the edits transforming the cognates from one language to the other ($\boldsymbol{\theta}_E, \boldsymbol{D}_E$).

The coding is redundant, as one language and the edits would be enough to reconstruct the second language. In the interest of symmetry between target languages, we ignore this redundancy.

The intuition is that the changes in spelling between the cognates in a particular language pair is regular. Coding the differences in a way that reduces the cost of making a similar change in another word guides the model towards learning these patterns from the data.

The coding of the edits is based on the Levenshtein (1966) algorithm. Let $(w^a, w^b)$ be a cognate pair and its current segmentation $\big((m_1^a, \ldots, m_n^a), (m_1^b, \ldots m_n^b)\big)$. The morphs are paired up sequentially. Note that the restrictions on the search algorithm guarantee that both segmentations contain the same number of morphs, $n$. For a morph pair $(m_i^a, m_i^b)$, the Levenshtein-minimal set of edits is calculated. Edits that are immediately adjacent to each other are merged. In order to improve the modeling of sound length change, we extend the edit in both languages to include the neighboring unchanged character, if one half of the edit is the empty string $\epsilon$, and the other contains another instance of character representing the sound being lengthened or shortened. This extension encodes a sound lengthening as e.g. ’a→aa’ instead of ’$\epsilon$ →a’. As the edits are cheaper to reuse once added to the edit lexicon, avoiding edits with $\epsilon$ on either side is beneficial to reduce spurious use. Finally, position information is discarded from the edits, leaving only the substrings, separated by a boundary symbol.

As an example, the edits found between *yhteenkuuluvuuspolitiikkaa* and *ühtekuuluvuspoliitika* are ’y→ü’, ’een→e’, ’uu→u’, ’ti→it’, and ’kka→k’.

The semi-supervised weighting scheme of Kohonen et al. (2010) can be applied to Cognate Morfessor. A new weighting parameter *edit_cost_weight* is added, and multiplicatively applied to both the lexicon and corpus costs of the edits.

The training algorithm is an iterative greedy local search very similar to the Morfessor Baseline algorithm. The algorithm finds an approximately minimizing solution to Eq 2. The recursive splitting algorithm from Morfessor Baseline is slightly modified. If a non-cognate is being reanalyzed, the normal algorithm is followed. Cognates are reanalyzed together. Recursive splitting is applied, with the restriction that if a morph in one language is split, then the corresponding cognate morph in the other language must be split as well. The Cartesian product of all combinations of valid split points for both languages is tried, and the pair of splits minimizing the cost function is selected, unless not splitting results in even lower cost.

## 3 Extracting cognates from parallel data

Finnish–Estonian cognates were automatically extracted from the shared task training data. As we needed a Finnish–Estonian parallel data set, we generated one by triangulation from the English–Finnish and English–Estonian parallel data. This resulted in a set of 679 252 sentence pairs (ca 12 million tokens per language).

FastAlign (Dyer et al., 2013) was used for word alignment in both directions, after which the alignments were symmetrized using the *grow-diag-final-and* heuristic. All aligned word pairs were extracted based on the symmetrized alignment. Words containing punctuation, and pairs aligned to each other fewer than 2 times were removed. The list of word pairs was filtered based on Levenshtein distance. If either of the words consisted of 4 or fewer characters, an exact match was required. Otherwise, a Levenshtein distance up to a third of the mean of the lengths, rounding up, was allowed. This procedure resulted in a list of 40 472 cognate pairs. The list contains words participating in multiple cognate pairs. Cognate Morfessor is only able to link a word to a single cognate. We filtered the list, keeping only the pairing to the most frequent cognate, which reduces the list to 22 226 pairs.

The word alignment provides a check for semantic similarity in the form of translational equivalence. Even though the word alignment may produce some errors, accidentally segmenting false friends consistently should not be problematic.

## 4 Data

After filtering, we have 9 million multilingual sentence pairs in total. 6.3M of this is English–Finnish, of which 2.2M is parallel data, and 4.1M is synthetic backtranslated data. Of the 2.8M total English–Estonian, 1M is parallel and 1.8M backtranslated. The sentences backtranslated from Finnish were from the news.2016.fi corpus, translated with a PBSMT model, trained with WMT16 constrained settings. The backtranslation from Estonian was freshly made with a BPE-based system similar to our baseline system, trained on the WMT18 data. The sentences were selected from the news.20{14-17}.et corpora, using a language model filtering technique.

### 4.1 Preprocessing

The preprocessing pipeline consisted of filtering by length[3] and ratio of lengths[4], fixing encoding problems, normalizing punctuation, removing of rare characters[5], deduplication, tokenizing, truecasing, rule-based filtering of noise, normalization of contractions, and filtering of noise using a language model.

The language model based noise filtering was performed by training a character-based deep LSTM language model on the in-domain monolingual data, using it to score each target sentence in the parallel data, and removal of sentences with perplexity per character above a manually picked threshold. A lenient threshold[6] was selected in order to filter noise, rather than for aiming for domain adaptation. The same process was applied to filter the Estonian news data for backtranslation.

Our cognate segmentation resulted in a target vocabulary of 42 386 subwords for Estonian and 46 930 subwords for Finnish, resulting in 64 396 subwords when combined.

For segmentation of the English source, a separate Morfessor Baseline model was trained. To ensure consistency between source and target segmentations, we used the segmentation of the Cognate Morfessor model for any English words that were also present in the target side corpora. The source vocabulary consisted of 61 644 subwords.

As a baseline segmentation, we train a shared 100k subword vocabulary using BPE. To produce a balanced multilingual segmentation, the following procedure was used: First, word counts were calculated individually for English and each of the target languages Finnish and Estonian. The counts were normalized to equalize the sum of the counts for each language. This avoided imbalance in the amount of data skewing the segmentation in favor of some language. BPE was trained on the balanced counts. Segmentation boundaries around hyphens were forced, overriding the BPE.

[3] 1–100 tokens, 3–600 chars, $\leq$ 50 chars/token.

[4] Requiring ratio 0.5–2.0, if either side $>$ 10 chars.

[5] $<$ 10 occurrences

[6] 96% of the data was retained.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\epsilon$ | → | n | 27919 | g | → | k | 3000 | il | → | $\epsilon$ | 2077 |
| $\epsilon$ | → | a | 17082 | ü | → | y | 2979 | m | → | mm | 2016 |
| $\epsilon$ | → | i | 15725 | oo | → | o | 2790 | s | → | n | 2005 |
| d | → | t | 12599 | t | → | a | 2674 | ee | → | e | 1950 |
| l | → | ll | 5236 | $\epsilon$ | → | k | 2583 | i | → | $\epsilon$ | 1889 |
| $\epsilon$ | → | ä | 4437 | aa | → | a | 2536 | $\epsilon$ | → | e | 1803 |
| s | → | ssa | 3907 | õ | → | o | 2493 | u | → | o | 1724 |
| t | → | tt | 3863 | a | → | ä | 2479 | $\epsilon$ | → | d | 1496 |
| o | → | u | 3768 | s | → | $\epsilon$ | 2173 | il | → | t | 1486 |
| e | → | i | 3182 | t | → | $\epsilon$ | 2158 | d | → | $\epsilon$ | 1433 |

Table 2: 30 most frequent edits learned by the model. The direction is Estonian→Finnish. The numbers indicate how many times the edit was applied in the morph lexicon. $\epsilon$ indicates the empty string.

| | chrF-1.0 | BLEU% |
|---|---|---|
| EN-ET | dev | dev |
| BPE | 56.52 | 17.93 |
| monolingual | 53.44 | 15.82 |
| Cognate Morfessor | 57.05 | 18.40 |
| +finetuned | 57.23 | 18.45 |
| +ensemble-of-5 | **57.75** | **19.09** |
| +ensemble-of-3 | 57.64 | 18.96 |
| +linked embeddings | 56.20 | 17.48 |
| −LM filtering | 52.94 | 14.65 |
| 6+6 layers | 57.35 | 18.84 |

Table 3: Development set results for English–Estonian. character-F and BLEU scores in percentages. +/− stands for adding/removing a component. Multiple modifications are indicated by increasing the indentation.

Multilingual translation with target-language tag was done following (Johnson et al., 2016). A pseudo-word, e.g. <TO_ET> to mark Estonian as the target language, was prefixed to each paired English source sentence.

## 5 NMT system

We use the OpenNMT-py (Klein et al., 2017) implementation of the Transformer.

### 5.1 Transformer

The Transformer architecture (Vaswani et al., 2017) relies fully on attention mechanisms, without need for recurrence or convolution. A Transformer is a deep stack of layers, consisting of two types of sub-layer: multi-head (MH) attention (Att) sub-layers and feed-forward (FF) sub-layers:

$$\begin{aligned}\mathrm{Att}(Q,K,V) &= \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \\ a_i &= \mathrm{Att}(QW_i^Q, KW_i^K, VW_i^V) \\ \mathrm{MH}(Q,K,V) &= [a_1; \ldots ; a_h]W^O \\ \mathrm{FF}(x) &= \max(0, xW_1 + b_1)W_2 + b_2\end{aligned} \tag{3}$$

where $Q$ is the input query, $K$ is the key, and $V$ the attended values. Each sub-layer is individually wrapped in a residual connection and layer normalization.

When used in translation, Transformer layers are stacked into an encoder-decoder structure. In the encoder, the layer consists of a self-attention sub-layer followed by a FF sub-layer. In self-attention, the output of the previous layer is used as queries, keys and values $Q = K = V$. In the decoder, a third context attention sub-layer is inserted between the self-attention and the FF. In context attention, $Q$ is again the output of the previous layer, but $K = V$ is the output of the encoder stack. The decoder self-attention is also masked to prevent access to future information. Sinusoidal position encoding makes word order information available.

### 5.2 Training

Based on some preliminary results, we decided to reduce the number of layers to 4 in both encoder and decoder; later we found that the decision was based on too short training time. Other parameters were chosen following the OpenNMT FAQ (Rush, 2018): 512-dimensional word embeddings and hidden states, dropout 0.1, batch size 4096 tokens, label smoothing 0.1, Adam with initial learning rate 2 and $\beta_2$ 0.998.

Fine-tuning for each target language was performed by continuing training of a multilingual model. Only the appropriate monolingual subset of the training data was used in this phase. The data was still prefixed for target language as during multilingual training. No vocabulary pruning was performed.

In our ensemble decoding procedure, the predictions of 3–8 models are combined by averaging after the softmax layer. Best results are achieved when the models have been independently trained. However, we also try combinations where a second copy of a model is further trained with a different configuration (monolingual finetuning).

|  | chrF-1.0 |  |  |  | BLEU% |  |  |  |
|---|---|---|---|---|---|---|---|---|
| EN-FI | nt2015 | nt2016 | nt2017 | nt2017AB | nt2015 | nt2016 | nt2017 | nt2017AB |
| BPE | 58.59 | 59.76 | 62.00 | 63.06 | 21.09 | 21.04 | 23.49 | 26.55 |
| monolingual | 57.94 | 59.11 | 61.33 | 62.41 | 20.87 | 20.70 | 23.11 | 26.12 |
| Cognate Morfessor | 58.18 | 59.81 | 62.15 | 63.24 | 20.73 | 21.18 | 23.37 | 26.26 |
| +finetuned | 58.48 | 59.89 | 62.17 | 63.28 | 21.08 | 21.41 | 23.45 | 26.52 |
| +ensemble-of-8 | **59.07** | **60.69** | **62.94** | **64.07** | **21.50** | **22.34** | **24.59** | **27.55** |
| −LM filtering | 58.19 | 59.39 | 61.78 | 62.82 | 20.62 | 20.77 | 23.38 | 26.36 |
| +linked embeddings | 57.79 | 59.45 | 61.52 | 62.58 | 19.95 | 20.84 | 22.70 | 25.69 |
| 6+6 layers | 58.68 | 60.26 | 62.37 | 63.52 | 21.05 | 21.81 | 23.93 | 27.08 |

Table 4: Results for English–Finnish. character-F and BLEU scores in percentages. +/− stands for adding/removing a component. Newstest is abbreviated nt. Both references are used in nt2017AB.

We experimented with partially linking the embeddings of cognate morphs. In this experiment, we used morph embeddings concatenated from two parts: a part consisting of normal embedding of the morph, and a part that was shared between both halves of the cognate morph pair. Non-cognate morphs used an unlinked embedding also for the second part. After concatenation, the linked embeddings have the same size as the baseline embeddings.

We evaluate the systems with cased BLEU using the mteval-v13a.pl script, and characterF (Popovic, 2015) with $\beta$ set to 1.0. The latter was used for tuning.

## 6 Results

Based on preliminary experiments, the Morfessor corpus cost weight $\alpha$ was set to 0.01, and the edit cost weight was set to 10. The most frequent edits are shown in Table 2.

Table 3 shows the development set results for Estonian. Table 4 shows results for previous year's test sets for Finnish.

The tables show our main system and the two baselines: a multilingual model using joint BPE segmentation, and a monolingual model using Morfessor Baseline.

Cognate Morfessor outperforms the comparable BPE system according to both measures for Estonian, and according to chrF-1.0 for Finnish. For Finnish, results measured with BLEU vary between test sets. The cross-lingual segmentation is particularly beneficial for Estonian.

In the monolingual experiment, the cross-lingual segmentations are replaced with monolingual Morfessor Baseline segmentation, and only the data sets of one language pair at a time is used. These results show that even the higher resourced language, Finnish, benefits from multilingual training.

The indented rows show variant configurations of our main system. Monolingual finetuning consistently improves results for both languages. For Estonian, we have two ensemble configurations: one combining 3 monolingually finetuned independent runs, and one combining 5 monolingually finetuned savepoints from 4 independent runs. Selection of savepoints for the ensemble was based on development set chrF-1. In the ensemble-of-5, one training run contributed two models: starting finetuning from epochs 14 and 21 of the multi-lingual training. The submitted system is the ensemble-of-3, as the ensemble-of-5 finished training after the deadline. For Finnish, we use an ensemble of 4 finetuned and 4 non-finetuned savepoints from 4 independent runs.

To see if further cross-lingual learning could be achieved, we performed an unsuccessful experiment with linked embeddings. It appears that explicit linking does not improve the morph representations over what the translation model is already capable of learning.

After the deadline, we trained a single model with 6 layers in both the encoder and decoder. This configuration consistently improves results compared to the submitted system.

All the variant configurations (ensemble, finetuning, LM filtering, linked embeddings, number of layers) used with Cognate Morfessor are compatible with each other. We did not not explore the combinations in this work, except for combining finetuning with ensembleing: all of the models in the Estonian ensembles, and 4 of the models in the Finnish ensemble are finetuned. All the variant configurations except for linked embeddings could also be used with BPE.

## 7 Conclusions and future work

The translation system trained using the Cognate Morfessor segmentation outperforms the baselines for both languages. The benefit is larger for Estonian, the language with less data in this experiment.

One downside is that, due to the model structure, Cognate Morfessor is currently not applicable to more than two target languages.

Cognate Morfessor itself learns to model the frequent edits between cognate pairs. However, in the preprocessing cognate extraction step of this work, we used unweighted Levenshtein distance, which does not distinguish edits by frequency. In future work, weighted or graphonological Levenshtein distance could be applied (Babych, 2016).

## Acknowledgments

This research has been supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 780069. Computer resources within the Aalto University School of Science "Science-IT" project were used. We wish to thank Peter Smit for groundlaying work that led to Cognate Morfessor.

## References

Bogdan Babych. 2016. Graphonological Levenshtein edit distance: Application for automated cognate identification. *Baltic Journal of Modern Computing*, 4(2):115–128.

Michael Bloodgood and Benjamin Strauss. 2017. Using global constraints and reranking to improve cognates detection. In *Proc. ACL*, volume 1, pages 1983–1992.

Alina Maria Ciobanu and Liviu P Dinu. 2014. Automatic detection of cognates using orthographic alignment. In *Proc. ACL*, volume 2, pages 99–105.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proc. SIGPHON*, pages 21–30, Philadelphia, PA, USA. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proc. NAACL*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Oskar Kohonen, Sami Virpioja, and Mikaela Klami. 2009. Allomorfessor: Towards unsupervised morpheme analysis. In *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 975–982. Springer Berlin / Heidelberg.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proc. SIGMORPHON*, pages 78–86, Uppsala, Sweden. Association for Computational Linguistics.

Grzegorz Kondrak. 2001. Identifying cognates by phonetic and semantic similarity. In *Proc. NAACL*, pages 1–8. Association for Computational Linguistics.

Grzegorz Kondrak. 2009. Identification of cognates and recurrent sound correspondences in word lists. *TAL*, 50(2):201–235.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Ruslan Mitkov, Viktor Pekar, Dimitar Blagoev, and Andrea Mulloni. 2007. Methods for extracting and classifying pairs of cognates and false friends. *Machine translation*, 21(1):29.

Jason Naradowsky and Kristina Toutanova. 2011. Unsupervised bilingual morpheme segmentation and alignment with context-rich hidden semi-Markov models. In *Proc. ACL: HLT*, pages 895–904. Association for Computational Linguistics.

Maja Popovic. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *WMT15*, pages 392–395.

Taraka Rama. 2016. Siamese convolutional networks for cognate identification. In *Proc. COLING*, pages 1018–1027.

Alexander M. Rush. 2018. OpenNMT FAQ – How do i use the Transformer model? http://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model. Accessed: 27.7.2018.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. In *Proc. ACL16*.

Benjamin Snyder and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proc. ACL: HLT*, pages 737–745.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. NIPS*, pages 6000–6010.

Sami Virpioja, Oskar Kohonen, and Krista Lagus. 2010. Unsupervised morpheme analysis with Allomorfessor. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, volume 6241 of *Lecture Notes in Computer Science*, pages 609–616. Springer Berlin / Heidelberg.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for Morfessor Baseline. Report 25/2013 in Aalto University publication series SCIENCE + TECHNOLOGY, Department of Signal Processing and Acoustics, Aalto University.

Richard Wicentowski. 2004. Multilingual noise-robust supervised morphological analysis using the WordFrame model. In *Proc. SIGPHON*, pages 70–77. Association for Computational Linguistics.

# The AFRL WMT18 Systems: Ensembling, Continuation and Combination

**Jeremy Gwinnup, Timothy Anderson**
**Grant Erdmann, Katherine Young**
Air Force Research Laboratory
{jeremy.gwinnup.1, timothy.anderson.20
grant.erdmann,katherine.young.1.ctr}
@us.af.mil

## Abstract

This paper describes the Air Force Research Laboratory (AFRL) machine translation systems and the improvements that were developed during the WMT18 evaluation campaign. This year, we examined the developments and additions to popular neural machine translation toolkits and measure improvements in performance on the Russian–English language pair.

## 1 Introduction

As part of the 2018 Conference on Machine Translation (Bojar et al., 2018) news-translation shared task, the AFRL human language technology team participated in the Russian–English portion of the competition. We largely employed our strategies from last year (Gwinnup et al., 2017), but adapted them to the past year's developments, including the University of Edinburgh's "bi-deep" (Miceli Barone et al., 2017; Sennrich et al., 2017) and Google's transformer (Vaswani et al., 2017) architectures. For Russian–English we again submitted an entry comprising our best systems trained with Marian (Junczys-Dowmunt et al., 2018), OpenNMT (Klein et al., 2017), and Moses (Koehn et al., 2007) combined using the Jane system combination method (Freitag et al., 2014).

## 2 Data and Preprocessing

We used and preprocess data as outlined in Gwinnup et al. (2017). For some systems, we included the Russian–English portion of the Paracrawl[1] corpus despite the noisy nature of the data. For all systems trained, we applied byte-pair encoding (BPE) (Sennrich et al., 2016) to address the vocabulary-size problem.

## 3 MT Systems

This year, we focused system-building efforts on the Marian, OpenNMT, and Moses toolkits, having explored a variety of parameters, data, and conditions.

### 3.1 Marian

We spent most of our effort investigating variations in our experimental setup with the Marian toolkit, varying training corpora, network architecture and validation metrics.

In order to facilitate ease of ensembling of models and to reduce variables while comparing the effects of settings with our Marian systems we held constant the following settings:

- We trained a joint BPE model with 49500 splits.
- We held the vocabulary size constant during training to 90k entries each for source and target.
- We held the word embedding dimensionality to 512 for all models.
- We used 1024 units in the hidden layer (where appropriate).
- We exclusively used `newstest2014` as the validation set.

We experimented with building both bi-deep and transformer models - we used the same network settings with each to again provide a basis for comparison between other conditions.

For the bi-deep systems we used the following parameters:

- Alternating encoder
- Encoder cell depth of 2

[1] http://www.paracrawl.eu

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 394–398
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64038

- Encoder layer depth of 4
- Decoder cell base depth of 4
- Decoder cell 'high depth' of 2
- Decoder layer depth of 4
- Layer normalization
- Tied embeddings for source, target and output layers
- Skip-connections

For the transformer models we used the following parameters:

- 6 layer encoder
- 6 layer decoder
- 8 transformer heads
- Tied embeddings for source, target and output layers
- Layer normalization
- Label smoothing
- Learning rate warm-up and cool-down

#### 3.1.1 Validation Metric Choice

We experimented with varying the metric used during training to determine if using an alternate metric yielded improvements. Based on comments from previous years' efforts, we employed BEER 2.0 (Stanojević and Sima'an, 2014) as an alternate validation metric. BEER is a trained machine translation evaluation metric with high correlation with human judgment both on sentence and corpus level. Use of this metric is motivated by the human evaluation portion of the WMT news translation task.

To compare this effect, we trained three bi-deep systems on the parallel corpus used in our WMT17 submission. These systems are trained with our common parameters outlined above, only varying the choice of validation metric: cross-entropy, BLEU, and BEER. The results of this comparison are shown in Table 1. We noted that cross-entropy and BLEU as validation metrics produce similar BLEU scores for the available test sets, but the use of BEER as a validation metric yielded an increase of between +0.7 and +1.5 BLEU when decoding the test sets.

#### 3.1.2 Pretrained Word Embeddings

Settling on the choice of BEER as a validation metric, we then investigated the use of pretrained word embeddings (Neishi et al., 2017) in order to boost translation performance. We took the Russian and English monolingual CommonCrawl (Smith et al., 2013) data provided by the organizers and applied tokenization and BPE with our common, joint model. We then used `word2vec` (Mikolov et al., 2013) to train word embeddings with 512 dimensions on each of the prepared corpora. These embeddings were then used during model training. We did not fix these word embeddings while training.

For comparison purposes, we trained a bi-deep model on the WMT18 provided training data, using our common criteria with BEER as a validation metric (as outlined in Section 3.1.1). The results of this comparison are shown in Table 2. We noted an over +1.0 BLEU improvement across all available test sets solely from the use of these pretrained word embeddings.

#### 3.1.3 Training Corpus Choice

The last major comparison for our Marian systems involved the choice of training corpora. For various training runs, we used the corpus from our WMT17 system, which included backtranslated data generated by a Marian 'Amun' system as described in Gwinnup et al. (2017). For others, we used the entirety of the WMT18 preprocessed data provided by the organizers. We trained bi-deep systems with pretrained word embeddings, with BEER as a validation metric, for both the WMT18 provided data and the concatenation of both the WMT17 and WMT18 corpora described earlier.

The results of this comparison are shown in Table 3. We noted there is between a +0.7 and +1.5 BLEU increase for test sets not used for validation purposes (`newstest2014` showed an increase of +2.1 BLEU, but this may be due to the models overfitting on the validation set.)

#### 3.1.4 Fine Tuning

We briefly examined fine-tuning (or continued training) (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016) late into the evaluation period. A fine-tuning corpus was constructed from the concatenation of all of the news task testsets from 2013 to 2017. A bi-deep model trained on both the WMT18 preprocessed data and the data used from our WMT17 system, pretrained word embeddings

| System | newstest2013 | newstest2014 | newstest2015 | newstest 2016 | newstest2017 |
|---|---|---|---|---|---|
| cross-entropy-valid | 24.60 | 30.32 | 26.98 | 26.78 | 27.71 |
| bleu-valid | 24.74 | 30.23 | 26.63 | 26.94 | 27.42 |
| beer-valid | 25.43 | 31.51 | 28.09 | 28.10 | 28.74 |

Table 1: Comparison between using cross-entropy, BLEU and BEER as validation metrics with Marian systems. Scores for various WMT test sets measured in cased BLEU.

| System | newstest2013 | newstest2014 | newstest2015 | newstest 2016 | newstest2017 |
|---|---|---|---|---|---|
| during training | 25.63 | 31.20 | – | 26.68 | 29.60 |
| pretrained | 26.72 | 32.59 | 28.69 | 28.41 | 31.56 |

Table 2: Comparison on using pretrained word embeddings with Marian systems. Scores for various WMT test sets measured in cased BLEU.

| Corpus | newstest2013 | newstest2014 | newstest2015 | newstest 2016 | newstest2017 |
|---|---|---|---|---|---|
| wmt17backtrans | 27.75 | 33.83 | 31.07 | 30.24 | 21.39 |
| wmt18preproc | 26.72 | 32.59 | 28.69 | 28.41 | 31.56 |
| wmt17/18 concat | 28.03 | 34.70 | 30.21 | 29.67 | 32.21 |

Table 3: Comparison of different training corpora conditions. Scores for various WMT test sets measured in cased BLEU.

and validated with BEER was chosen as a starting point. We use the fine-tuning corpus to continue training for only two epochs. The results of this comparison are shown in Table 4. A gain of almost +3 BLEU is observed, showing promise with this technique, however concerns arise over possible overfitting to the fine-tuning corpus.

| System | BLEU | BEER |
|---|---|---|
| general | 27.05 | 0.575 |
| fine-tuned | 30.02 | 0.597 |

Table 4: Standard and Fine-tune results for `newstest2018` measured in cased BLEU and BEER.

#### 3.1.5 Marian Submission System

We ultimately employed an ensemble system of 5 bi-deep models and 6 transformer models trained in varying conditions (with the exception of the finetuned system in Section 3.1.4) outlined above as the Marian contribution to our submission system. This system also employed a R2L transformer model performing rescoring on the n-best lists generated during the decoding step.

### 3.2 OpenNMT

Our OpenNMT system trained on the provided parallel data excepting paracrawl and the back-translated corpus we employed for our WMT17 system. This system uses a standard RNN architecture and was fine-tuned with the other available news task test sets.

All systems used 1000 hidden units and 600 unit word embeddings.

### 3.3 Moses

In order to provide diversity for system combination, we trained a phrase-based Moses (Koehn et al., 2007) system with the same data as the Marian system outlined in Section 3.1. This system employed a hierarchical reordering model (Galley and Manning, 2008) and 5-gram operation sequence model (Durrani et al., 2011). The 5-gram English language model was trained with KenLM on the constrained monolingual corpus from our WMT15 (Gwinnup et al., 2015) efforts. The BPE model used was applied to both the parallel training data and the language modeling corpus. System weights were tuned with the Drem (Erdmann and Gwinnup, 2015) optimizer using the “Expected Corpus BLEU” (ECB) metric.

## 4 System Combination

Jane System combination (Freitag et al., 2014) was employed to combine outputs from the best systems from each approach outlined above. Individual component system and final combination scores are shown in Table 5. The final system combination output comprised our entry to the Russian–English portion of the WMT18 news task evaluation.

| System | BLEU | BEER |
|---|---|---|
| Marian | 29.42 | 0.592 |
| OpenNMT | 28.88 | 0.580 |
| Moses | 24.25 | 0.565 |
| Syscomb | 30.01 | 0.597 |

Table 5: System combination and input system scores measured in BLEU and BEER on the `newstest2018` test set.

## 5 Conclusion

We presented a series of improvements to our Russian–English systems focusing on improvements to neural machine translation toolkits. We again combined the best of several approaches via system combination creating a composite submission exhibiting the best of all contributing approaches.

## References

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics (ACL '11)*, pages 1045–1054, Portland, Oregon.

Grant Erdmann and Jeremy Gwinnup. 2015. Drem: The AFRL submission to the WMT15 tuning task. In *Proc. of the Tenth Workshop on Statistical Machine Translation*, pages 422–427, Lisbon, Portugal.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast Domain Adaptation for Neural Machine Translation. *CoRR*, abs/1612.06897.

Markus Freitag, Matthias Huck, and Hermann Ney. 2014. Jane: Open source machine translation system combination. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32, Gothenburg, Sweden.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 848–856.

Jeremy Gwinnup, Tim Anderson, Grant Erdmann, Katherine Young, Christina May, Michaeel Kazi, Elizabeth Salesky, and Brian Thompson. 2015. The afrl-mitll wmt15 system: There's more than one way to decode it! In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 112–119, Lisbon, Portugal. Association for Computational Linguistics.

Jeremy Gwinnup, Timothy Anderson, Grant Erdmann, Katherine Young, Michaeel Kazi, Elizabeth Salesky, Brian Thompson, and Jonathan Taylor. 2017. The afrl-mitll wmt17 systems: Old, new, borrowed, bleu. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 303–309, Copenhagen, Denmark. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, Melbourne, Australia.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford Neural Machine Translation Systems for Spoken Language Domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

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 16 Aug 2018. Originator reference number RH-18-118721. Case number 88ABW-2018-4047.

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep architectures for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 99–107. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *International Conference on Learning Representations (ICLR) Workshop*.

Masato Neishi, Jin Sakuma, Satoshi Tohda, Shonosuke Ishiwatari, Naoki Yoshinaga, and Masashi Toyoda. 2017. A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 99–109. Asian Federation of Natural Language Processing.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Jason R. Smith, Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL '13)*, pages 1374–1383, Sofia, Bulgaria.

Miloš Stanojević and Khalil Sima'an. 2014. Fitting sentence level translation evaluation with many dense features. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 202–206, Doha, Qatar. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

# The University of Edinburgh's Submissions to the WMT18 News Translation Task

**Barry Haddow, Nikolay Bogoychev, Denis Emelin, Ulrich Germann,**
**Roman Grundkiewicz, Kenneth Heafield, Antonio Valerio Miceli Barone** and **Rico Sennrich**
University of Edinburgh, Scotland

## Abstract

The University of Edinburgh made submissions to all 14 language pairs in the news translation task, with strong performances in most pairs. We introduce new RNN-variant, mixed RNN/Transformer ensembles, data selection and weighting, and extensions to back-translation.

## 1 Introduction

For the WMT18 news translation task, we were the only team to make submissions to all 14 language pairs. Our submissions built on our strong results of the WMT16 and WMT17 tasks (Sennrich et al., 2016a, 2017), in that we used neural machine translation (NMT) with byte-pair encoding (BPE) (Sennrich et al., 2016c), back-translation (Sennrich et al., 2016b) and deep RNNs (Miceli Barone et al., 2017). For this year's submissions we experimented with new architectures, and new ways of data handling. In brief, the innovations that we introduced this year are:

**Architecture** This year we experimented with the Transformer architecture (Vaswani et al., 2017), as implemented by Marian (Junczys-Dowmunt et al., 2018), as well as introducing a new variant on the deep RNN architectire (Section 2.3).

**Data selection and weighting** For some language pairs, we experimented with different data selection schemes, motivated by the introduction of the noisy ParaCrawl corpora to the task (Section 2.1). We also applied weighting of different corpora to most language pairs, particularly DE↔EN (Section 3.5).

**Extensions to Back-translation** For TR↔EN (Section 3.7) we used copied monolingual data (Currey et al., 2017a) and iterative back-translation.

**In-domain Fine-tuning** For RU↔EN (Section 3.6) we fine-tuned using a specially constructed "in-domain" data set.

## 2 System Details

In this section we describe the general properties of our systems, as well as some novel approaches that we tried this year such as data selection and a variant on the GRU-based RNN architecture. The specifics of our submissions for each language pair are described in Section 3.

### 2.1 Data and Selection

All our systems were constrained in the sense that they only used the supplied parallel data (including ParaCrawl) for training the systems. We also used the monolingual news crawls to create extra synthetic parallel data by back-translation, for all language pairs, and by copying monolingual data for TR↔EN. During training we generally used *newsdev2016* or *newstest2016* for validation, and *newstest2017* for development testing (i.e. model selection), except for ZH↔EN, and ET↔EN, where we used the recent *newsdev* sets instead.

All parallel data contains a certain amount of noise, and the problem was exacerbated this year since the organisers provided a ParaCrawl corpus[1] for most language pairs[2] as additional training data. On inspection, we could see that these crawled corpora were quite noisy, including mis-aligned sentence pairs, incorrect language, and garbled encodings. In early experiments, we showed increases in BLEU from including ParaCrawl in the training data, for ET→EN and FI→EN, but we decided to see if we could improve performance further by applying data filtering. We experimented with different filtering methods, described below.

[1] `https://paracrawl.eu`

[2] ParaCrawl corpora was not available for EN↔TR and EN↔ZH.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 399–409
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64039

**Language Identifier Filtering** This was applied to the CS↔EN and DE↔EN corpora, based on observations that CzEng, and ParaCrawl both contain sentence pairs in the "wrong" language. For CS↔EN we applied langid (Lui and Baldwin, 2012) to both sids of the data, removing any sentences whose English side is not labelled as English, or whose Czech is not labelled as Czech, Slovak or Slovenian[3]. For DE↔EN, we just applied langid to ParaCrawl and retained only those pairs where each side was identified as the 'correct' language by *langid*. This reduced the size of the ParaCrawl corpus from about 36 million sentence pairs to ca. 18 million sentence pairs.

**Data Selection with Translation Perplexity** We applied this to ET↔EN and FI↔EN. To perform the filtering, we first trained shallow RNN models in both directions, using all the permitted parallel data except ParaCrawl. We then used these models to score the ParaCrawl sentence pairs, normalising by target sentence length, and adding the scores for forward and reverse models. We then ranked sentence pairs in ParaCrawl using this score, and performed a grid search across different thresholds (from 0 – 100% in 10 point intervals) of the ParaCrawl data, in addition to the other parallel data. We trained a shallow RNN system using the data selected across each of these thresholds, and tested it on *newstest2017* (for FI→EN), or half of *newsdev2018* (for ET→EN).

The results of the filtering are shown in Figure 1. Based on these results, we chose a threshold of 0.3 for ET↔EN (which gives us +0.8 BLEU), but used the whole of ParaCrawl for FI→EN.

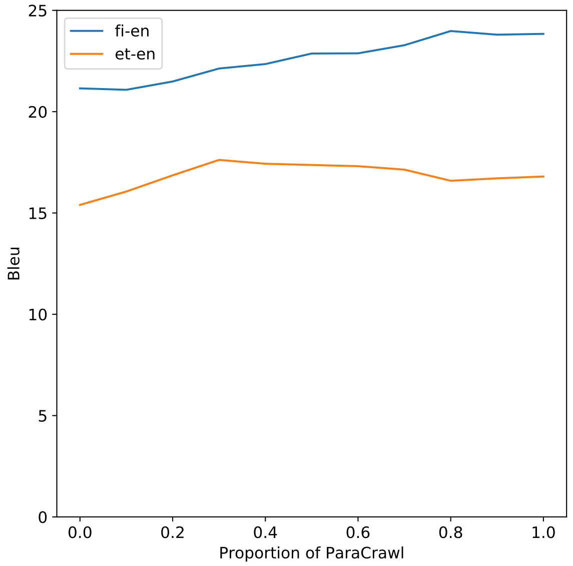


Figure 1: Result of translation perplexity filtering of ParaCrawl on 2 language pairs

**Alignment-based Filtering** We applied this to the DE→EN parallel data, after langid filtering. We word-aligned all pre-cleaned parallel data with *fastalign* (Dyer et al., 2013) and computed the geometric mean of forward and backward alignment probabilities as a coarse estimate of how good a translation pair the respective sentence pair is.

All parallel data was sorted in descending order of this "plausible translation" score, and a neural system was trained on this data, in this order. In order to determine a threshold for data filtering, we monitored the performance on a validation set (*newstest2016*) and observed the point where translation quality started to deteriorate. We used the translation plausibility score at this point as the threshold for selecting data for training the final systems.

## 2.2 Preprocessing

For most language pairs, our preprocessing setup consisted of the Moses pipeline (Koehn et al., 2007) of normalisation, tokenisation and truecasing, followed by byte-pair encoding (BPE) (Sennrich et al., 2016c). We generally applied joint BPE, with the number of merge operations set on a per-pair basis, detailed in Section 3. Different pipelines were used for processing the two languages written in non-Latin scripts (i.e. Chinese and Russian), also explained in Section 3. For some language pairs (those including Czech, Estonian, Finnish and German) we used the preprocessed data provided by the organisers (which is preprocessed up to truecasing), whilst for the others we started with the raw data.

## 2.3 Model Architecture

For this submission we considered two types of sequence-to-sequence architectures: a transformer (Vaswani et al., 2017) and a deep RNN, specifically the BiDeep GRU encoder-decoder (Miceli Bar-

[3] langid identified a significant proportion of the data as these other two Slavic languages, but on inspecting a sample, they were found nearly always to be Czech. The issue with langid is that we just give it the text, without providing any prior knowledge, when in actual fact there is a strong prior that CzEng sentences are really Czech and English, by construction

one et al., 2017). Both architectures[4] are implemented in the Marian open source neural machine translation framework (Junczys-Dowmunt et al., 2018). For the transformer architecture we used the “wmt2017-transformer” setup from the Marian example collection[5].

We extended the RNN with with multi-head and multi-hop attention. Multi-head attention is similar to Chen et al. (2018), with an MLP attention mechanism using a single $\tanh$ hidden layer followed by one soft-max layer for each attention heads. We further include an optional projection layer on the attended context with layer normalisation in order to avoid increasing the total size of the attended context.

Let $C \in \mathcal{R}^{N_s \times d_e}$ be the input sentence representation produced by the encoder, where $N_s$ is the source sentence length and $d_e$ is the top-level bidirectional encoder state dimension. Let $s \in \mathcal{R}^{d_d}$ be an internal decoder state at some step. Then for source sentence position $i$ we compute a vector of $M$ attention weights, where $M$ is the number of attention heads:

$$\begin{aligned} W, A &\in \mathcal{R}^{N_s \times M} \\ W_i &= \text{MLP}(C_i, s) \\ A_i &= \frac{\exp(W_i)}{\sum_{i'} \exp(W_{i'})} \end{aligned}$$

where we assume that exponentiation is applied element-wise. Then we compute the attended context vector as:

$$\text{ATT}(C, h) = \text{CAT}_{r=1}^{M} \left( \sum_i \text{PROJ}_r(C_i) \cdot a_{i,r} \right)$$

where $CAT_{r=1}^{M}$ is vector concatenation over the attention heads and each $PROJ_r$ is either the identity function or a trainable linear layer followed by layer normalization.

Multi-hop attention is similar to Gehring et al. (2017), except that we do not use convolutional layers, but instead we introduce additional attention hops between the layers of the deep transition GRU in the decoder. In our implementation multi-head and multi-hop attention can be combined, in which case each attention hop is a separate multi-head attention mechanism.

Let $L_t \geq 2$ be the decoder base recurrence depth and $H < L_t$ be the number of attention hops. Then the base level of the decoder is defined as:

$$\begin{aligned} s_{j,1} = &\text{GRU}_1 \left(y_{j-1}, s_{j-1,L_t}\right) \\ s_{j,k} = &\text{GRU}_k \left(\text{ATT}_k(C, s_{j,k-1}), s_{j,k-1}\right) \\ &\text{for } 1 < k \leq H + 1 \\ s_{j,k} = &\text{GRU}_k \left(0, s_{j,k-1}\right) \\ &\text{for } H + 1 < k \leq L_t \end{aligned}$$

where each $\text{ATT}_k(C, s)$ is and independent multi-head attention mechanism with $M$ heads. For a BiDeep decoder, the higher levels are the same as in the default Marian implementation of the BiDeep architecture [6].

### 2.4 Training

All our systems are trained with Marian[7] (Junczys-Dowmunt et al., 2018), using Adam (Kingma and Ba, 2015). To improve training stability and generalisation, we employed label smoothing (0.1) (Szegedy et al., 2016), exponential smoothing (i.e. Polyak averaging) with 0.0001 weight, gradient clipping and layer normalisation (Ba et al., 2016). For all pairs except CS↔EN (where it harmed BLEU) we used dropout (Srivastava et al., 2014; Gal and Ghahramani, 2016) on the Transformer/RNN connections.

## 3 Submitted Systems

### 3.1 Chinese ↔ English

For ZH↔EN we preprocessed the parallel data, which consists of NewsCommentary v13, UN data and CWMT, as follows. We first desegmented all the Chinese data and resegmented it using Jieba[8]. We then removed any sentences that did not contain Chinese characters on the Chinese side, or contained only Chinese characters on the English side. We also cleaned up all sentences containing links, sentences longer than 50 words, as well as sentences where the amount of tokens on either side was $>$ 1.3 times the tokens on the other side, following Hassan et al. (2018). After preprocessing the corpus size was 23.6M sentences. We then applied BPE using 18,000 merge operations and we used the top 18,000 BPE segments as vocabulary. We augmented our data with backtranslated

[4] The BiDeep GRU is obtainable using the `-best-deep` option.

[5] https://github.com/marian-nmt/marian-examples

[6] The implementation of the multi-head and multi-hop attention architectures is available at: https://github.com/EdinburghNLP/marian-dev

[7] https://marian-nmt.github.io

[8] https://github.com/fxsjy/jieba

ZH↔EN from Sennrich et al. (2017), which consists of 8.6M sentences for EN→ZH and 19.7M for ZH→EN.

We trained using the BiDeep architecture with multi-head attention with 1 hop and 3 heads. We decoded using an ensemble of 5 L2R systems and a beam of 12 for EN→ZH and 6 L2R systems and a beam of 12 for ZH→EN. Due to time constraints, we were not able to train any of the systems to convergence.

### 3.2 Czech ↔ English

After preprocessing, language filtering (see Sections 2.1 and 2.2), and removing any parallel sentences where neither side contains an ASCII letter, we were left with around 50M sentence pairs. We then learned a joint BPE model over the source and target corpora, with 89,500 merge operations, and applied it using a vocabulary threshold of 50.

For back-translation, we trained shallow RNN models in both directions without ParaCrawl or the langid-based corpus cleaning, and used to decode with a beam size of 5. We back-translated the English 2017 news-crawl, and the Czech news-crawls from 2016 and 2017, removing lines with more than 50 tokens, to create additional corpora of approximately 26.5M sentence for CS→EN and 13M for EN→CS. Initially we tried simply concatenating each of these corpora with the natural parallel data, but this gave poor results for CS→EN, so we over-sampled the synthetic data 2 times for that pair to give approximately equal amounts for synthetic and natural data. For EN→CS, we did not see any benefit from equalising the synthetic/natural ratio, so we stuck to using simple concatenation.

For the submitted systems, we trained the BiDeep RNN models using Marian. In addition to the default Marian settings, we used layer normalisation, tied embeddings, label smoothing (0.1), exponential smoothing, no dropout, but we used multiheaded/multihop attention with 2 heads and 3 hops. We trained on 4 GPUs with a working memory of 4000MB on each, validating every 2,500 updates. We used exponential smoothing and took the final smoothed model. We trained 4 left-right (L2R) and 4 right-left (R2L) models for each language pair, and due to time constraints we did not train to convergence, stopping each run after about 250k–350k updates. We decoded using an ensemble of the 4 L2R systems and a beam of 50, then reranked with the 4 R2L systems. For both language pairs we normalised probabilities by target length, raising it to a power of 0.8 for CS→EN.

### 3.3 Estonian ↔ English

As explained in Section 2.1, we used a filtered ParaCrawl for this pair, and in common with CS↔EN we removed any sentence pairs where either side contained no ascii letter. We trained and applied a BPE model with 89,500 merge operations and a vocabulary threshold of 50. We split *newsdev2018* randomly and used one half for validation and another half for development testing.

The models used for back-translation were shallow RNNs trained on the parallel data without ParaCrawl. We translated the 2017 English news-crawl to Estonian, and translated all the Estonian news-crawls to English. We also experimented with the BigEst Estonian corpus, but did not see any improvement when using it to produce synthetic data, nor when we selected 50% of it using Moore-Lewis selection (Moore and Lewis, 2010) with the news-crawl data as in-domain. Our final natural parallel corpus contains approximately 1.2M sentences, and the synthetic corpora are about 2.9M for EN→ET and 26.5M for ET→EN. To create the final corpora for training, we combined natural and synthetic, over-sampling the natural 3-times for EN→ET and 23-times for the ET→EN. Again we apply BPE, trained on the Europarl, Rapid and selected Paracrawl corpora, with the same parameters as before.

Our submitted system was an ensemble of 4 left-right systems, reranked with 4 right-left systems, with each ensemble consisting of 2 deep BiDeep RNNs and 2 Transformers. The RNN had a BiDeep architecture, with layer normalisation, tied embeddings, label smoothing (0.1), exponential smoothing, RNN dropout (0.2), source and target word dropout (0.1) and multihead/multihop attention with 2 heads and 3 hops. We trained on 4 GPUs with a working memory of 4000MB on each, validating every 2,500 updates. The RNNs were not trained to convergence (due to time constraints) but stopped after between 300k and 500k steps. The transformer models used the settings from Marian examples. without layer normalisation, with a working memory of 9500MB (on each of 4 GPUs), validating every 2500 updates, and detecting convergence with a patience of 10. We also applied source and target word dropout to the transformer models. They generally converged in under 200k updates. As for CS↔EN we used exponentially smoothed models. Decoding is the same as for CS↔EN, with normalisation by target length.

### 3.4 Finnish ↔ English

For FI↔EN, after pre-processing we removed sentence pairs where either side contains no ascii characters, then trained and applied a BPE model with 89,500 merge operations and a vocabulary threshold of 50. As reported in Section 2.1, we used the whole of ParaCrawl in our system.

For back-translation, we trained shallow RNN models in each direction, without ParaCrawl. We back-translated with a beam size of 5, translating the English 2017 news-crawl to Finnish, and the Finnish 2014–2017 news-crawls to English. Before back-translation, we removed any sentences with length greater than 50 tokens. For EN→FI, we combined 3.2M naturally parallel sentence pairs, over-sampling 5-times, with 14.6M sentences of synthetic data. For FI→EN, we combined the same natural corpus (over-sampled 8-times) with a 26.5M corpus of synthetic parallel data.

We created the submitted systems in the same way as the ET↔EN systems (Section 3.3), and again we were not able to train the deep RNNs to convergence. The only difference is that for EN→FI, we normalise by the target length raised to a power of $0.5$, after running a grid search over different normalisations on the development set.

### 3.5 German ↔ English

Our efforts focussed on extracting the most useful data from ParaCrawl. After preprocessing and selection (see Section 2.1, we trained and applied joint BPE models with 35,000 merge operations, and a threshold of 50.

To balance the data, we blended the data in a mix as shown in Table 1, by randomly sampling from each corpus (without raplacement), resetting (i.e., replacing all items at once) each corpus when it became exhausted, for a total of 40 million sentence pairs.

Our system was based on the transformer in Marian examples, and initially we trained several left-right and right-left systems with tied target embeddings (but separate source embeddings). We used these systems to create ensembles.

For the translation direction EN→DE, we also trained a single model with a set-up more closely reflecting the setup described in the *wmt2017-transformer* Marian example set-up. For this single decoder, we tied all embeddings and pooled the top-ranked 7.5 million sentence pairs from paracrawl (according to the translation plausibility score) with the other training data. Below, this system is referred to as *single transformer*.

| Corpus | % |
|---|---|
| Back translations[1] | 50% |
| CommonCrawl | 5% |
| Europarl | 15% |
| News-commentary | 10% |
| ParaCrawl | 10% |
| Rapid | 10% |

Table 1: Blend of data for training the DE↔EN ensemble models (40M sentence pairs total).

For the single transformer we used a mix of approximately 4.6 million parallel sentence pairs from latest versions of Europarl, CommonCrawl and News-commentary, oversampled twice, the 7.5 million parallel sentence pairs from ParaCrawl, filtered as described above, and 10 million back-translated sentences from NewsCrawl 2016. We trained a Marian transformer model with standard settings.

We also ran preliminary experiments with multi-head and multi-hop GRU architectures on the same training data except ParaCrawl but we found that these models tended to underperform the transformer by $0.6 - 1.0$ BLEU points, therefore we did not use them for our submission.

As the results in Table 2 show, the single transformer produces better results than our ensembles. Even re-ranking of the single transformer output deteriorates the results, which we attribute to lower quality of the models used for ensembling and re-ranking. At this point we do not know whether the differences in model quality are due to differences in the tying of parameters, different choices of other hyperparameters, differences in the training data used, or a combination of any of these potential causes.

### 3.6 Russian ↔ English

After preprocessing, we trained a joint BPE model with 90,000 merge operations, using the same Latin-Cyrillic transliteration trick as in Sennrich et al. (2016c). For back-translation we trained a deep RNN and translated Russian news crawls

| Pair | System | BLEU |
|---|---|---|
| DE→EN | Ensemble of 3 L2R, reranked with ensemble of 2 R2L | 43.9 |
| EN→DE | Single transformer | 44.4 |
| | Single transformer, reranked with ensemble of 2 R2L | 43.2 |
| | Ensemble of 2 L2R, reranked with ensemble of 2 R2L | 41.8 |

Table 2: WMT18 Results for German ↔ English

from 2015–2017, and the English news crawl from 2017 to give about 36M sentences in each direction.

In order to maximize the performance of our submission systems, we created a pseudo "in-domain" fine-tuning corpus designed to be representative of the targeted news domain to a greater extent than the full parallel corpus. For that purpose, we concatenated pre-processed sentence pairs from NewsCommentary v13, CommonCrawl, and Yandex Corpus, excluding the noisy ParaCrawl data as well as data from the UN Parallel Corpus V1.0 which has little overlap with our target domain. To ensure that the so assembled corpus is as free of noise as possible, we furthermore filter out sentence pairs in which the Russian side is not predominantly composed of Cyrillic characters or the English side is dominated by non-Latin characters. Lastly, we combined the so obtained "in-domain" corpus with an equal amount of back-translated news data, resulting in two datasets of 2.1M sentence pairs each.

Our final submission included both deep RNN models (using multi-head and multi-hop attention with 3 heads and 2 hops) and Transformer models similar to the Transformer-Base of Vaswani et al. (2017). For the RNNs, we applied layer normalisation, label smoothing (0.1), dropout between recurrent layers (0.1), exponential smoothing and tieall embeddings. We applied similar options to our transformer models.

We trained our models in two stages: 1) Training on the full parallel corpus and 2) Fine-tuning on the "in-domain" corpus with a reduced learning rate. Each of the submitted models was optimized using the Adam algorithm, with $\beta_1$ set to 0.9, and $\beta_2$ set to 0.98. Learning rate was set to 0.0003 during the training stage and lowered to 0.00003 during the fine-tuning stage. Throughout the training, the learning rate was linearly increased over the initial 16,000 update steps up to the specified value and gradually degraded thereafter.

Model validation was performed every 5,000 steps, and we terminated training if no BLEU improvements are observed after five consecutive validations. For fine-tuning, we initialized our models with parameters corresponding to the highest validation-BLEU on the full corpus and train until convergence, as indicated by early stopping, on the "fine-tuning" training set. Due to time-constaints, convergence could not be reached for several of the ensembled models.

Our final submissions consisted of an ensemble of 4 deep RNNs for EN→RU and a mixed ensemble of 2 RNNs and 2 transformers for RU→EN. All these models were trained independently and fine-tuned on the "in-domain" set. Improvements obtained following the fine-tuning step are detailed in Table 3. While our original intention was to use mixed ensembles for both directions, our transformer models under-performed on the EN→RU translation task, which we assume is due to our hyper-parameter choices. We re-ranked the translations obtained by our left-right ensemble with a right-left ensemble of identical design. It should be noted, however, that we were unable to identify any significant improvements in terms of validation-BLEU as a result of the re-ranking. We also fine-tuned the beam-size and length penalty hyper-parameters of our ensemble systems on the corresponding validation sets for which we observe a small increase in validation-BLEU. Accordingly, we set the beam size to 20 and length normalisation parameter to 0.4 for our EN→RU ensemble and to 28 and 1.2 respectively for RU→EN.

### 3.7 Turkish ↔ English

After preprocessing we trained and applied a joint BPE model with 36,000 merge operations, discarding any sentences longer than 120 tokens. To produce back-translations we built systems in two steps: first we trained back-translation systems in both directions using the parallel data only, and then we re-trained them on data sets containing additional 800K back-translated sentences. Back-

| | Deep RNN | | | Transformer | | |
|---|---|---|---|---|---|---|
| Direction | base | fine-tuned | significance | base | fine-tuned | significance |
| EN→RU | 30.25 | 32.69 | $p < 0.00001$ | - | - | - |
| RU→EN | 35.79 | 36.5 | $p < 0.005$ | 35.81 | 36.96 | $p < 0.005$ |

Table 3: Impact of in-domain fine-tuning on the RU ↔ EN task. Reported are best validation-BLEU scores averaged over all single models of the denoted type in the submitted ensemble systems. Statistical significance was established using a paired, two-tailed t-test.

| Corpus | # Synth. | R | # Total |
|---|---|---|---|
| A | 800K | ×1 | 1M |
| B | 2.5M | ×5 | 3.5M |
| C | 2.5M + 1M | ×5 | 4.5M |

Table 4: Training data sets for TR↔EN systems. Data sets consist of back-translated and original parallel data oversampled R times.

translation systems are trained as deep RNN models described below. The final training sets consist of 2.5M of synthetic parallel sentence pairs created from English or Turkish NewsCrawl data sets and the SETIMES2 data oversampled 5 times (Table 4). We also experimented with copying monolingual data (Currey et al., 2017b) by adding additional 1M examples with source sentences identical to target sentences randomly selected from the monolingual data.

Our RNN models used the BiDeep architecture, and we augmented the models with layer normalisation, skip connections, and parameter tying between all embeddings and output layer. The RNN hidden state size was set to 1024, embeddings size to 512.

The architecture of transformer models was close to the Transformer-Base proposed by Vaswani et al. (2017): encoder and decoder were composed of 6 layers, and employed 8-head self-attention. We used dropout between transformer layers (0.2) as well as in attention (0.05) and feed-forward layers (0.05). The rest of parameters remained the same as in the RNN models.

Optimization used 4 GPUs with synchronous training and mini-batch size fitted into 9.5GB of GPU memory. The learning rate was linearly increased to 0.0004 reaching this value after first 18,000 updates, and then decreased by a square of the passed updates starting at 24,000 update. As a stopping criterium we used early stopping with a patience of 10 based on the word-level cross-entropy on the *newsdev2016* data sets, which served as a development set. The model was validated every 5,000 updates, and we kept best models according to the cross-entropy and BLEU score.

We evaluated systems using models with the highest BLEU score on the development set. Decoding was performed by beam search with a beam size of 12 with length normalisation with value 0.2 for EN→TR and 1.2 for TR→EN based on the greed search on the development set. Additionally, as the Turkish language is not supported by the Moses tokenizer falling back to general English tokenization rules resulting in suboptimal detokenization, we postprocessed translated Turkish texts by merging words that contains an apostrophe.

We report results on the *newstest2017* and *newstest2018* in Table 5[9]. Our first submitted TR↔EN systems were ensembles of 6 independently trained models, reranked with 3 right-left systems (Ensemble ×6 +Rerank R2L ×3). Ensembles consist of four models trained on corpus B and one model trained on corpora A and C, while each right-left model is trained on different corpora A-C. Our final systems extended the previous ensemble by 6 additional models from the same training runs that achieve best cross-entropy (instead of best BLEU) on the development set[10], utilizing 12 left-right models in total (Ensemble ×6×2). For comparison, we report the results for single systems trained on different corpora, and there is no significant performance difference among them.

### 3.8 Overall Performance of Submissions

In Table 6 we show the BLEU scores of our systems as compared to the top-scoring constrained systems, giving the BLEU scores from the matrix[11] and the

[9] For our official submissions we also used n-best lists generated with the beam size of 20 instead of 30, which may explain the difference between the official and reported BLEU scores.

[10] Models achieving best cross-entropy differ from the models with highest BLEU for each training run.

[11] http://matrix.statmt.org

| System | EN-TR 2017 | EN-TR 2018 | TR-EN 2017 | TR-EN 2018 |
|---|---|---|---|---|
| Deep RNN$_A$ | 22.0 | 18.1 | 23.8 | 24.4 |
| Deep RNN$_B$ | 22.1 | 18.6 | 23.9 | 25.1 |
| Transformer$_A$ | 23.4 | 19.1 | 24.6 | 25.8 |
| Transformer$_B$ | 23.1 | 19.2 | 25.0 | 26.7 |
| Transformer$_C$ | 22.8 | 19.0 | 25.2 | 26.7 |
| +Ensemble ×6 | 24.0 | 19.9 | 26.2 | 27.6 |
| +Rerank R2L ×3 | 24.4 | 19.9 | 26.6 | 28.2 |
| +Ensemble ×6×2 | 24.3 | 19.9 | 26.3 | 27.7 |
| +Rerank R2L ×3 | 24.7 | 20.1 | 26.5 | 28.1 |
| Submission | | 19.5 | | 26.9 |

Table 5: Results for EN↔TR systems on official WMT test sets.

human evaluation from the findings paper (Bojar et al., 2018).

In terms of the clustering provided by the organisers, we were in the top constrained cluster (i.e. no significant difference was observed between ours and the best constrained system) for EN→CS, DE→EN, ET→EN, FI→EN, TR→EN and EN→TR, i.e. 6/14 language pairs. Nevertheless, Table 6 still shows that our systems generally lag behind the best submitted systems. This is contrast to the 2017 shared task, where we achieved the highest scores in most of the language pairs where we submitted systems. We hypothesise that other groups have taken fuller advantage of the transformer architecture, and also of data weighting and selection. We also suggest that covering all 14 language pairs meant that we had insufficient time for experimentation on some pairs, and in fact we were not able to train all models to convergence.

## 4 Post-Submission Experiments

In this section we present results of some post-submission experiments, which attempted to provide more insight into the contribution of different features of our system. We were especially interested in understanding why our systems tended to lag behind the performance of the best systems (in BLEU, at least). Mostly the experiments were conducted on EN↔{CS,ET,FI}.

The results are given on *newstest2017* (devtest) and *newstest2018* (test), except for ET↔EN, where devtest is half of *newsdev2018*.

### 4.1 Effect of Multihead/Multihop Attention

In the deep RNN models in our submissions, we used the BiDeep architecture, with multi-head/multihop attention, setting the number of hops to 3 and heads to 2. In Table 7, we show the effect of this on 3 different language pairs (both directions). For these experiments, we use the same training sets and data preparation as in our system submissions, but train the deep RNNs with a working memory of 10GB, validating every 1,000 steps, and testing for convergence with a patience of 10. We use exponential smoothing and show the results on a single smoothed model.

From the results in Table 7 we see that the multi-head/hop extension has a small positive effect on BLEU in most language pairs.

### 4.2 Effect of Vocabulary Size

After looking at the submission results, we questioned whether smaller vocabularies would have given better results, especially for transformer models. Having smaller vocabularies means that the models have few parameters, and also allow more words to be fitted into each training mini-batch.

To create a model with a smaller vocabulary, we follow the preparation steps used for our submissions (in EN↔{CS,ET,FI}), but use 30,000 BPE merges instead of 89,500. We show the effect both on the deep RNN model and on the Transformer model, and additionally we show the effect of tying all embeddings (i.e. source, target input and target output) on the Transformer model. The submitted models for these language pairs only have the target input and output embeddings tied. As in Section 4.1 we set the working memory for the deep RNN to 10GB, and we set the working memory for transformer training to 9.5GB. We used layer normalisation for the transformer models (although this appeared to make little if any difference to the results). In Table 8 we show the comparison for RNNs, and in Table 9 we show the same comparison for Transformer models.

Examining the results in Tables 8 and 9, we can see that the effect of vocabulary size reduction on RNN models is mixed, whereas the transformer models have a preference (in BLEU, at least) for smaller vocabularies. Tying all embeddings does not seem to help. Further investigation is needed on the vocabulary size question though, as the relationship between BPE hyper-parameters and BLEU is unclear. We note that changes in the vocabulary

|  | X→ EN |  |  |  | EN→X |  |  |  |
|---|---|---|---|---|---|---|---|---|
|  | Ours | Top | Δ BLEU | Δ DA | Ours | Top | Δ BLEU | Δ DA |
| CS | 31.8 | 33.9 | -3.13 | -3.9 | 23.4 | 26.0 | -2.59 | -6.6 |
| DE | 43.9 | 48.4 | -4.49 | -4.5 | 44.4 | 48.3 | -3.95 | -5.6 |
| ET | 29.4 | 30.7 | -1.30 | -1.9 | 22.7 | 23.6 | -0.85 | -4.6 |
| FI | 23.5 | 24.9 | -1.40 | -1.2 | 16.7 | 18.2 | -1.53 | -5.5 |
| RU | 32.8 | 34.9 | -2.12 | -3.5 | 29.8 | 34.8 | -4.95 | -6.0 |
| TR | 26.9 | 28.0 | -1.10 | -1.1 | 19.5 | 20.0 | -0.48 | 0.0 |
| ZH | 24.0 | 29.3 | -5.31 | -4.3 | 33.3 | 43.8 | -10.5 | -10.0 |

Table 6: Overall BLEU scores of our systems, compared to the top-scoring constrained systems. We also show the difference with the direct assessment (DA) score of the best constrained system.

|  | No hop/head |  | 3 hop, 2 head |  |
|---|---|---|---|---|
| Pair | devtest | test | devtest | test |
| CS-EN | 30.0 | 30.8 | **30.6** | **31.2** |
| EN-CS | 23.2 | 23.0 | **23.6** | **23.2** |
| ET-EN | 24.8 | **27.9** | **25.4** | 27.2 |
| EN-ET | **18.9** | **21.6** | 18.8 | 21.1 |
| FI-EN | 31.4 | 22.6 | **31.9** | **23.1** |
| EN-FI | 24.4 | 16.0 | **25.2** | **16.2** |

Table 7: Comparison of performance of deep RNN models with/without the multihop/multihead extension.

|  | BPE 89.5k |  | BPE 30k |  |
|---|---|---|---|---|
| Pair | devtest | test | devtest | test |
| CS-EN | 30.6 | **31.2** | **30.8** | 31.1 |
| EN-CS | **23.6** | **23.2** | 23.0 | 22.9 |
| ET-EN | 25.4 | 27.2 | **26.1** | **28.2** |
| EN-ET | **18.8** | 21.1 | 18.7 | 21.1 |
| FI-EN | **31.9** | **23.1** | 31.6 | 22.8 |
| EN-FI | 25.2 | 16.2 | **25.5** | **16.5** |

Table 8: Effect of reducing vocabulary size for deep RNN models. We used 89,500 BPE merges for our submissions, but tried reducing it to 30,000 for post-submission experiments.

size could have a disproportionate effect on the translation of rare words (including proper nouns) which would not necessarily be detected by BLEU.

### 4.3 Mixed Ensembles

For our submitted systems for FI↔EN and ET↔EN we used mixed ensembles consisting of two deep RNNs and two Transformer models. In this section we examine whether the mix of architectures in the ensemble is beneficial. We compare this mixed ensemble with an ensemble of four deep RNNs.

In Table 10, we show the results. We show the mean BLEU score of the models in the ensemble, together with the overal ensemble score. For clarity, we just show scores on our test set (*newstest2018*). The gain in BLEU from ensembling (over the mean BLEU) is slightly higher in all cases than the corresponding gain for the uniform ensemble.

## 5 Conclusions

We have described Edinburgh's systems for all 14 language pairs, showing that we can gain improvements by augmenting a GRU-based RNN with multi-head and multi-hop attention, using mixed ensembles of deep RNNs and transformers, and selecting data from the noisy ParaCrawl corpora. Our systems perform strongly in most language pairs, except for when we did not manage to train to convergence.

## Acknowledgments

This work was supported in part by funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 688139 (SUMMA), and from the Connecting Europe Facility under agreement No. NEA/CEF/ICT/A2016/1331648 (ParaCrawl).

Nikolay Bogoychev was funded by an Amazon faculty research award to Adam Lopez.

Computing resources were provided by:

- The Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service `http://www.csd3.cam.ac.uk/`,

| | BPE 89.5k, tied | | BPE 30k, tied | | BPE 30k, tied-all | |
|---|---|---|---|---|---|---|
| Pair | devtest | test | devtest | test | devtest | test |
| CS-EN | 28.9 | 29.4 | **29.4** | **29.8** | 29.1 | 29.5 |
| EN-CS | 22.6 | 22.4 | 22.8 | 22.5 | **23.2** | **22.6** |
| ET-EN | 25.4 | 27.8 | 24.9 | **27.9** | **25.8** | 27.8 |
| EN-ET | 18.8 | 21.3 | **19.4** | **21.9** | 19.3 | 21.8 |
| FI-EN | 30.7 | 22.2 | **31.3** | **22.3** | 30.7 | 22.2 |
| EN-FI | 24.8 | 16.2 | **25.9** | 16.6 | 25.6 | **16.7** |

Table 9: Effect of reducing vocabulary size for Transformer models. We used 89,500 BPE merges for our submissions, but tried reducing it to 30,000 for post-submission experiments. We also show the effect of tying all embeddings.

| | Mixed ensemble | | | Uniform ensemble | | |
|---|---|---|---|---|---|---|
| Pair | mean | ensemble | Δ | mean | ensemble | Δ |
| ET-EN | 27.7 | 29.0 | +1.3 | 27.5 | 28.6 | +1.1 |
| EN-ET | 21.5 | 22.7 | +1.2 | 21.0 | 21.9 | +0.9 |
| FI-EN | 22.5 | 23.2 | +0.7 | 22.1 | 22.7 | +0.6 |
| EN-FI | 15.9 | 16.5 | +0.7 | 15.6 | 16.0 | +0.4 |

Table 10: Effect of mixed versus uniform ensembles. The ensembles are either 2 deep RNNs and 2 transformers, or 4 RNNs.

provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk) `https://www.csd3.cam.ac.uk/getting-help/www.dirac.ac.uk%29`

- The Alan Turing Institute under the EPSRC grant EP/N510129/1.

## References

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *CoRR*, abs/1607.06450.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, and Philipp Koehn. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86. Association for Computational Linguistics.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017a. Copied Monolingual Data Improves Low-Resource Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017b. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, Will Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in c++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *The International Conference on Learning Representations*, San Diego, California, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the ACL-2007 Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep Architectures for Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference Short Papers*.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's Neural MT Systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, Copenhagen, Denmark.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 368–373, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

# TencentFmRD Neural Machine Translation for WMT18

**Bojie Hu**[1], **Ambyer Han**[2], **Shen Huang**[1]
[1] Tencent Research, Beijing, China
[2] Natural Language Processing Lab, Northeastern University, China
{bojiehu, springhuang}@tencent.com, ambyerhan0301@outlook.com

## Abstract

This paper describes the Neural Machine Translation (NMT) system of TencentFmRD for Chinese↔English news translation tasks of WMT 2018. Our systems are neural machine translation systems trained with our original system TenTrans. TenTrans is an improved NMT system based on Transformer self-attention mechanism. In addition to the basic settings of Transformer training, TenTrans uses multi-model fusion techniques, multiple features reranking, different segmentation models and joint learning. Finally, we adopt some data selection strategies to fine-tune the trained system and achieve a stable performance improvement. Our Chinese→English system achieved the second best BLEU scores and fourth best cased BLEU scores among all WMT18 submitted systems.

## 1 Introduction

End-to-end neural machine translation (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015) based on self-attention mechanism (Vaswani et al., 2017), the Transformer, has become promising paradigm in field of machine translation academia and industry. Experiments show Transformer, which does not rely on any convolutional or recurrent networks, to be superior in translation performance while being more parallelizable and requiring significantly less time to train. The training part of this paper is an improvement on the tensor2tensor[1] open source project based on the Transformer architecture, and the inference part is completely original, and we called this system TenTrans. We participated in two directions of translation tasks: English→Chinese and Chinese→English.

We divide TenTrans system into three parts to introduce in this paper. First, we introduce how to train better translation model, that is, the training phase. Second, we describe how good models can generate better translation candidates, that is, the inference phase. Finally, we describe $N$-best rescoring phase, which ensures that translation results which are closer to the expression typically produced by users are chosen. Our experimental setup is based on recent promising techniques in NMT, including using Byte Pair Encoding (BPE) (Sennrich et al., 2016b) and mixed word/character segmentation rather than words as modeling units to achieve open-vocabulary translation (Luong and Manning, 2016), using back-translation (Sennrich et al., 2016a) method and joint training (Zhang et al., 2018) applied to make use of monolingual data to enhance training data. And we also improve the performance using an ensemble based on six variants of the same network, which are trained with different parameter settings.

In addition, we design multi-dimensional features for strategic integration to select the best candidate from $n$-best translation lists. Then we perform minimum error rate training (MERT) (Och, 2003) on validation set to give different features corresponding reasonable weights. And we process named entities, such as person name, location name and organization name into generalization types in order to improve the performance of unknown named entity translation (Wang et al., 2017). Finally, we adopt some data selection strategies (Li et al., 2016) to fine-tune the trained system and achieve a stable performance improvement.

Our Chinese→English system achieved the second best BLEU (Papineni et al., 2002) scores and fourth best cased BLEU scores among all WMT18 submitted systems. The remainder of this paper is organized as follows: Section 2 describes the system architecture of TenTrans. Section 3 states

[1] https://github.com/tensorflow/tensor2tensor

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 410–417
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64040

all experimental techniques used in WMT18 news translation tasks. Sections 4 shows designed features for reranking $n$-best lists. Section 5 shows experimental settings and results. Finally, we conclude in section 6.

## 2 System Architecture of TenTrans

In this work, TenTrans has the same overall architecture as the Transformer: that is, it uses stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. The encoder and decoder both are composed of a stack of $N = 6$ identical layers. Each layer has two sub-layers, multi-head self-attention mechanism and position-wise connected feed-forward network. We add a residual connection (He et al., 2016) around each of the two sub-layers, followed by layer normalization (Ba et al., 2016). The left part of training phase in Figure 1 describes the structure of the basic sub-layer in the encoder and decoder. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. In this work we employ $multi\text{-}head = 16$ heads, that is, parallel attention layers.

For all our models, we adopt Adam (Kingma and Ba, 2015) ($\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$) as optimizer. We use model hidden state dimension 1024, the same as input embedding dimension and output embedding dimension. We linearly increase the learning rate whose initial value is 0.1 in the first $warmup = 6000$ training steps, and then anneal with the same way as Transformer. We use synchronous mini-batch SGD (Dean et al., 2012) training with $batch\_size = 6144$ and data parallelism on 8 NVIDIA Tesla P40 GPUs. We clip the gradient norm to 1.0 (Pascanu et al., 2013). We apply residual dropout (Zaremba et al., 2014; Srivastava et al., 2014) with $P_{rd} = 0.3$ to avoid overfitting. In training, we don't just focus on the word with highest probability score, but let the likelihood calculation be smoother, so applying label smoothing (Szegedy et al., 2016) with $\epsilon_{ls} = 0.1$. All weight parameters are initialized according to uniform distribution in the interval $[-0.08, 0.08]$. We will early stop training (Sennrich et al., 2017) when there is no new maximum value of the validation BLEU for 10 consecutive save-points (saving every 10000 updates) and select the model with the highest BLEU score on the validation set.

We mainly optimize TenTrans system through three parts. First, through the first part of Figure 1, multiple models are trained, and then the data selection method is used to continue to fine tune the system. Then, through the second part of Figure 1, the combination of best multiple models is used to decode the monolingual corpus to generate pseudo-bilingual data, and then the pseudo-bilingual data is proportionally added to the training set to continue the training of the first stage, and these two phases are continuously iterated until convergence. Finally, the third stage, $N$-best rescoring phase, finds the best translation result among the translation candidates by designed multiple sets of features. In order to learn the corresponding weights of multiple sets of features, the optimization is carried out through minimum error rate training (MERT).

## 3 Experimental Techniques

This section mainly introduces the techniques used in training and inference phase.

### 3.1 Multi-model Fusion Technology

For multi-model fusion, we try three strategies:

**Checkpoint ensembles (CE)**, refers to the last $N$ checkpoints saved during a single model training, where $N$ is set to 10. In addition, we add the best 10 models saved during the early stopping training.

**Independent parameter ensembles (IPE)**, refers to firstly training $N$ models with different initialization parameters, and then weighting the average probability distribution of multiple models when softmax layer is calculated. Here we set $N$ to 6, and we make better models have relatively higher weights, and poorer models have relatively lower weights.

**Independent model ensembles (IME)**. An independent model ensemble is a set of models, each of which has been assigned a weight. It is not necessary to perform calculating the probability distribution in the inference process. Our experimental results show that this method performs slightly lower than IPE method, but the advantage is that the decoding speed is the same as the single model decoding.

In this work, we use the checkpoint ensemble method to initially integrate each single model, and then use the independent parameter ensemble method to perform multi-model integration in the

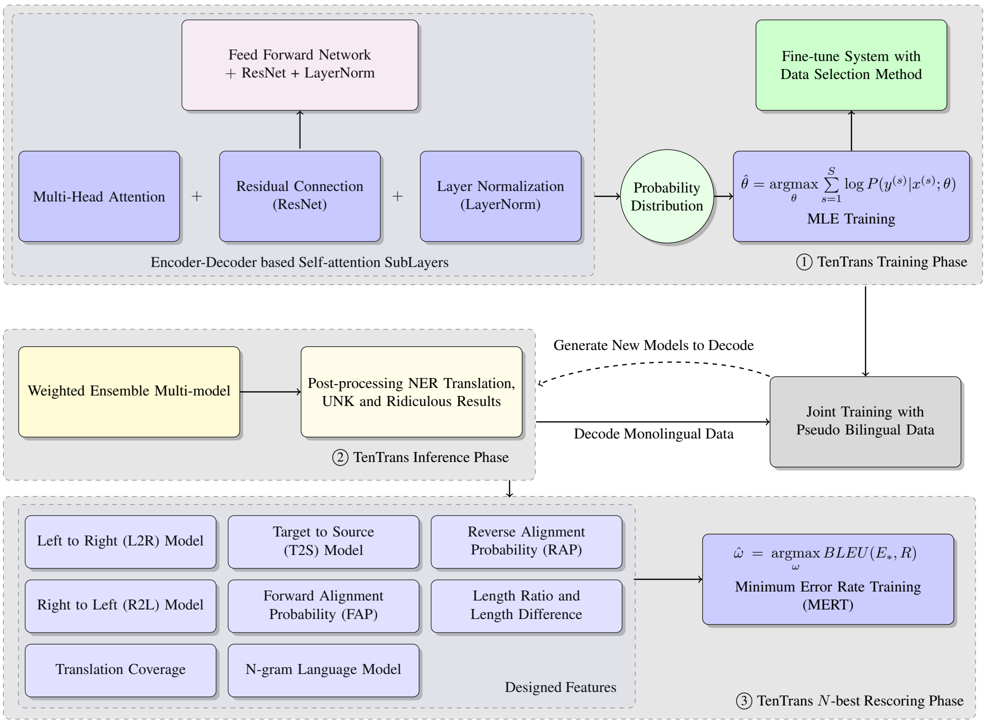

Feed Forward Network + ResNet + LayerNorm
Multi-Head Attention
+
Residual Connection (ResNet)
+
Layer Normalization (LayerNorm)
Encoder-Decoder based Self-attention SubLayers
Probability Distribution
$\hat{\theta} = \operatorname{argmax}_{\theta} \sum_{s=1}^{S} \log P(y^{(s)}|x^{(s)};\theta)$
MLE Training
Fine-tune System with Data Selection Method
① TenTrans Training Phase
Weighted Ensemble Multi-model
Post-processing NER Translation, UNK and Ridiculous Results
② TenTrans Inference Phase
Generate New Models to Decode
Decode Monolingual Data
Joint Training with Pseudo Bilingual Data
Left to Right (L2R) Model
Target to Source (T2S) Model
Reverse Alignment Probability (RAP)
Right to Left (R2L) Model
Forward Alignment Probability (FAP)
Length Ratio and Length Difference
Translation Coverage
N-gram Language Model
Designed Features
$\hat{\omega} = \operatorname{argmax}_{\omega} BLEU(E_*, R)$
Minimum Error Rate Training (MERT)
③ TenTrans $N$-best Rescoring Phase

Figure 1: An illustration of system architecture of TenTrans. $\theta$ indicates model parameters being trained, and $s$ indicates a training sample containing a source language $x$ and a target language $y$. $\omega$ are the feature weight parameters being tuned by MERT. Ridiculous results mainly refer to translation results that are extremely long or short and clearly inconsistent with the source language.

stage of generating the final result of the system. The independent model ensemble method is used to decode monolingual corpus to generate pseudo-bilingual data during joint training.

### 3.2 Fine-tune System with Data Selection Method

In mainstream machine translation systems, network parameters are fixed after the training is finished. The same model will be applied to various test sentences. A very important problem with this approach is that it is difficult for a model to self-adapt to different sentences, especially when there is a big difference between the test set field and training set field. To alleviate this problem, (Li et al., 2016) proposed to search similar sentences in the training data using the test sentence as a query, and then fine tune the NMT model using the retrieved training sentences for translating the test sentence. We follow the strategy of (Li et al., 2016). This method firstly learns the general model from the entire training corpus. Then, for each test sentence, we extract a small subset from the training data, consisting of sentence pairs whose source sides are similar to the testing sentence. The subset is used to fine tune the general model and get a specific model for every sentence. To calculate similarity between two sentences, we adopt Levenshtein distance, which calculates the minimum steps for converting a string to another string using insertion, deletion and substitution these operations. We firstly filter the training corpus by only considering those which have common words with the testing sentence, and then compute similarity with the filtered set. In order to speed up the calculation, we use the inverted index method.

### 3.3 Joint Training

This work uses the monolingual corpus to enhance the training set by joint training. Joint training refers to the use of the corresponding additional target side and source side monolingual data at the source-to-target (S2T) and the target-to-source (T2S) translation model, and jointly optimizing the two translation models through an iterative process. In each iteration, T2S model is used to generate pseudo bilingual data for S2T with target-side monolingual data, and S2T model is used to generate pseudo bilingual data for T2S

**Algorithm 1** Joint Training Algorithm in TenTrans System

```
Input: original bilingual data $B$, source monolingual data $X_m$, target monolingual data $Y_m$
Output: trained S2T models $M^i_{s2t}(i = 1 \cdots 6)$ and T2S models $M^i_{t2s}(i = 1 \cdots 6)$
Train 6 $M^i_{s2t}(i = 1 \cdots 6)$ and 6 $M^i_{t2s}(i = 1 \cdots 6)$ with different parameters
$n \Leftarrow 1$
while Not Converged do
    Integrate 6 $M^i_{s2t}(i = 1 \cdots 6)$ to generate $M^{ens}_{s2t}$ with IME method
    Integrate 6 $M^i_{t2s}(i = 1 \cdots 6)$ to generate $M^{ens}_{t2s}$ with IME method
    Use $M^{ens}_{t2s}$ to generate pseudo-training data $F_{t2s}$ by translating $Y_m$
    Use $M^{ens}_{s2t}$ to generate pseudo-training data $F_{s2t}$ by translating $X_m$
    New corpus to train $M^i_{s2t}(i = 1 \cdots 6)$, $C_{s2t} \Leftarrow n \times B + F_{t2s}$
    New corpus to train $M^i_{t2s}(i = 1 \cdots 6)$, $C_{t2s} \Leftarrow n \times B + F_{s2t}$
    $n \Leftarrow n + 1$
    Train $M^i_{s2t}$ with $L(\theta_{s2t}) = \sum_{s=1}^{S} \log P(y^{(s)}|x^{(s)}) + \sum_{t=1}^{T} \log P(y^{(t)}|x^{(t)}) P^{'}(x^{(t)}|y^{(t)})$ using $C_{s2t}$ [2]
    Train $M^i_{t2s}$ with $L(\theta_{t2s}) = \sum_{s=1}^{S} \log P(x^{(s)}|y^{(s)}) + \sum_{t=1}^{T} \log P(x^{(t)}|y^{(t)}) P^{'}(y^{(t)}|x^{(t)})$ using $C_{t2s}$ [2]
end while
```

with source-side monolingual data. This joint optimization approach enables the translation model in both directions to be improved, and generating better pseudo-training data to be added to the training set. Therefore, in the next iteration, it can train better T2S model and S2T model, so on and so forth. The right part of the decoding phase of Figure 1 outlines the iterative process of joint training. In addition, in order to solve the problem that back-translation often generates pseudo data with poor translation quality and thus affects model training, the generated training sentence pairs are weighted so that the negative impact of noisy translations can be minimized in joint training. Original bilingual sentence pairs are all weighted as 1, while the synthetic sentence pairs are weighted as the normalized corresponding model output probability. For the specific practice of joint training in this paper, see Algorithm 1.[2]

### 3.4 Different Modeling Units

We use BPE[3] with 50K operations in both source side and target side of Chinese→English translation. In English→Chinese translation task, we use BPE with 50K operations in English source side, and use mixed word/character segmentation in Chinese target side. We keep the most frequent 60K Chinese words and split other words into characters. In post-processing step, we simply remove all the spaces.

### 3.5 NER Generalization Method

To alleviate poor translation performance of named entities, we first use the pre-defined tags to replace named entities in training set to train a tagged NMT system, for example, use $\$number$ for numbers, $\$time$ for time, $\$date$ for date, $\$psn$ for person name, $\$loc$ for location name, $\$org$ for organization name. Then the key to the problem is how to identity these entities and classify them into corresponding types accurately. In order to solve this problem, we classify these entities into two types, one type that can be identified by rules, and the other type that can be identified by classification models. To decide whether an entity is a time, a number, or a date, we use finite automata (FA) (Thatcher and Wright, 1968). Aiming at the names of people, location names, and organization names, we first use biLSTM-CRF[4] (Lample et al., 2016; Huang et al., 2015) to train a Chinese sequence tagging model on ”People's Daily 1998” data set and an English sequence tagging model on CoNLL2003 data set, and then identify named entities at the source and target language side of the training set.

[2] Here $P^{'}(x^{(t)}|y^{(t)})$ refers to translation probability of $M^{ens}_{t2s}$ translating monolingual sentence $y^{(t)}$ to generate $x^{(t)}$, $P^{'}(y^{(t)}|x^{(t)})$ refers to translation probability of $M^{ens}_{s2t}$ translating monolingual sentence $x^{(t)}$ to generate $y^{(t)}$, $P(y^{(s)}|x^{(s)})$ denotes translation probability of $x^{(s)} \rightarrow y^{(s)}$ during training S2T model, and $P(x^{(s)}|y^{(s)})$ denotes translation probability of $y^{(s)} \rightarrow x^{(s)}$ during training T2S model.

[3] https://github.com/rsennrich/subword-nmt

[4] https://github.com/guillaumegenthial/sequence_tagging

In the test phase, we first convert these entities in the test sentences into corresponding predefined tags, and then directly using the tagged NMT system to translate the sentences. When a tag is generated at target side, we select the corresponding translation of the word in the source language side that has the highest alignment probability based attention probability with the same as tag type in target side. If the source side does not have the same type of tag, delete the current tag directly. In order to obtain the corresponding translation of each entity vocabulary, we obtain it in the phrase extraction stage in statistical machine translation (SMT) (Koehn, 2009). We extract a phrase pair with one source word number from phrase table, and then use target side of the phrase pair with highest frequency of occurrence as the translation of the word to construct a bilingual translation dictionary. Although this method has not greatly improved the BLEU evaluation metric, it is of great benefit to the readability of the translation result for human. We use UNK to represent out-of-vocabulary (OOV) words, and translate it in the same way as above.

## 4 Experimental Features

This section focuses on the features designed to help choose translation results which are closer to the way normal user expressions - that is, it focuses on the $N$-best rescoring phase. Several features designed in this work can be seen in the left part of third phase in Figure 1.

### 4.1 Right to Left (R2L) Model

Since the current translation models all carry out modeling from left to right, there is a tendency for the prefix part of translation candidates to be of higher quality than the suffix part (Liu et al., 2016). In order to alleviate this problem of translation imbalance, we adopt a right-to-left translation modeling method. Two R2L modeling method are used in this work: the first is that only the target data is inversed, and the second is that both the target data and the source data are inversed. Then, two models, R2L model and R2L-both model were trained. Finally, we also reverse the $n$-best lists and calculate the likelihood probability of each translation candidate given the source sentence using these two models. Each model mentioned above is integrated by training 6 models with different parameters.

### 4.2 Target to Source Model

Neural machine translation models often have the phenomena of missing translation, repeated translation, and obvious translation deviation (Tu et al., 2017). In order to alleviate this problem, we use the target-to-source translation system to reconstruct the source-to-target translation results to the source sentence. This approach can make it very difficult to reproduce poor translation results to the source sentence, and the corresponding probability score will be low. Similarly, these models are all integrated by multiple models.

### 4.3 Alignment Probability

In order to express the degree of mutual translation between the translation candidate and source sentence at the lexical level, the lexical alignment probability feature is adopted. This paper uses two kinds of alignment probabilities, forward alignment probability and backward alignment probability. The forward alignment probability indicates the degree of alignment of source language vocabulary to the target language vocabulary, while the backward alignment probability indicates the degree of alignment of target language vocabulary to the source language vocabulary. We obtain the alignment score by *fast-align* toolkit[5] (Dyer et al., 2013).

### 4.4 Length Ratio and Length Difference

In order to reflect the length ratio between source sentence and translation candidates, we designed the length ratio feature $R_{len} = Len(source)/Len(candidate)$ and the length difference feature $D_{len} = Len(source) - Len(candidate)$.

### 4.5 Translation Coverage

To reflect whether words in the source language sentences have been translated, we introduce translation coverage feature. This feature is calculated by adding one to the feature value if the source language words has been translated. We use the *fast-align* toolkit to count the top 50 target words with highest probability of aligning each source language word as the translation set of this source word.

[5] https://github.com/clab/fast_align

| System | C2E | E2C |
|---|---|---|
| baseline | 23.32 | 33.06 |
| +CE (checkpoint ensemble) | 24.06 | 33.84 |
| +IPE | 25.98 | 35.58 |
| +back-translation | 26.49 | 36.0 |
| +joint training | 26.96 | 36.51 |
| +fine-tune | 27.63 | 37.29 |
| +NER gereralization | 27.74 | 37.43 |
| +reranking (beam size 12) | 29.72 | 39.03 |
| +reranking (beam size 100) | 30.13 | 39.49 |
| **submitted system** | 30.21 | 39.61 |

Table 1: Chinese↔English BLEU results on WMT18 validation set. The ”C” and ”E” denotes Chinese and English respectively.

### 4.6 $N$-gram Language Model

For English, the word-level 5-gram language model is trained on the mixing corpus of ”News Crawl: articles from 2016” selected by newsdev2018 and English side of the training data. For Chinese, the character-level 5-gram language model is trained on the XMU. This work uses KenLM[6] toolkit (Heafield, 2011) to train $n$-gram language model.

### 4.7 Minimum Error Rate Training (MERT)

Obviously, some of the above features may be very powerful, while some of the effects are not particularly obvious. Therefore, we need to give each feature a corresponding weight. Our optimization goal is to find a set of feature weights that make the model score of translation candidates higher and the corresponding BLEU score higher. Therefore, we use minimum error rate training method to learn the feature weights

$$\begin{aligned} \omega^* &= \underset{\omega}{\operatorname{argmin}}\, Error(E_*, R) \\ &= \underset{\omega}{\operatorname{argmin}}\, (1 - BLEU(E_*, R)) \\ &= \underset{\omega}{\operatorname{argmax}}\, BLEU(E_*, R) \end{aligned} \tag{1}$$

where $\omega^*$ indicates tuned weights, $E_*$ indicates the best translation candidate for the source language and $R$ represents the corresponding reference translation.

## 5 Experimental Settings and Results

In all experiments, we report case-sensitive and detokenized BLEU using the NIST BLEU scorer. For Chinese output, we split to characters using the script supplied for WMT18 before running BLEU. In training and decoding phase, the Chinese sentences are segmented using NiuTrans (Xiao et al., 2012) Segmenter. For English sentences, we use the Moses (Koehn et al., 2007) tokenizer[7].

We used all the training data of WMT2018 Chinese↔English Translation tasks, firstly filtering out bilingual sentences with unrecognizable code, large length ratio difference, duplications and wrong language coding, then filtering out bilingual sentences with poor mutual translation rate by using *fast-align* toolkit. After data cleaning, 18.5 million sentence pairs remained. We used beam search with a beam size of 12, length penalty $\alpha = 0.8$ for Chinese→English system and length penalty $\alpha = 1.0$ for English→Chinese system. In order to recover the case information, we use Moses toolkit to train SMT-based recaser on English corpus. In addition, we also use some simple rules to restore the case information of the results. The size of the Chinese vocabulary and English vocabulary is 64k and 50k respectively after BPE operation. Table 1 shows the Chinese↔English translation results on development set of WMT2018. Wherein reranking refers to multi-feature based rescore method mentioned above. The submitted system in Table 1 has slightly better performance than is seen in the previous experiment because we have manually written some rules. As can be seen from the Table 1, when we increase the size of $n$-best from 12 to 100, the performance is improved by 0.41 BLEU after reranking based on multiple features.

## 6 Conclusion

In training phase of TenTrans, we report five experimental techniques. In the rescoring phase, we designed multiple features to ensure that candidates which are more likely to be produced by users are as close as possible to the top of $n$-best lists. Finally, our Chinese→English system achieved the second best BLEU scores among all WMT18 submitted systems.

[6] https://github.com/kpu/kenlm

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

## References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. 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)*, pages 1724–1734. Association for Computational Linguistics.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. 2012. Large scale distributed deep networks. In *Advances in neural information processing systems*, pages 1223–1231.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations, San Diego, California, United States*.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2016. One sentence one model for neural machine translation. *arXiv preprint arXiv:1609.06490*.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Franz Josef Och. 2003. 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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's Neural MT Systems for WMT17. *arXiv preprint arXiv:1708.00726*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

James W. Thatcher and Jesse B. Wright. 1968. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical systems theory*, 2(1):57–81.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *AAAI*, pages 3097–3103.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017. Sogou neural machine translation systems for wmt17. In *Proceedings of the Second Conference on Machine Translation*, pages 410–415.

Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. Niutrans: an open source toolkit for phrase-based and syntax-based machine translation. In *Proceedings of the ACL 2012 System Demonstrations*, pages 19–24. Association for Computational Linguistics.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. *arXiv preprint*, arXiv:1803.00353.

# The MLLP-UPV German-English Machine Translation System for WMT18

**Javier Iranzo-Sánchez, Pau Baquero-Arnal, Gonçal V. Garcés Díaz-Munío,**
**Adrià Martínez-Villaronga, Jorge Civera, Alfons Juan**
Machine Learning and Language Processing research group
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n, 46022, València, Spain
{jiranzo,pbaquero,ggarces,amartinez1,jcivera,ajuan}@dsic.upv.es

## Abstract

This paper describes the statistical machine translation system built by the MLLP research group of Universitat Politècnica de València for the German→English news translation shared task of the EMNLP 2018 Third Conference on Machine Translation (WMT18). We used an ensemble of Transformer architecture–based neural machine translation systems. To train our system under "constrained" conditions, we filtered the provided parallel data with a scoring technique using character-based language models, and we added parallel data based on synthetic source sentences generated from the provided monolingual corpora.

## 1 Introduction

In this paper we describe the statistical machine translation (SMT) system built by the MLLP research group of Universitat Politècnica de València for the German→English news translation shared task of the EMNLP 2018 Third Conference on Machine Translation (WMT18).

Neural Machine Translation (NMT) has made great advances over the last few years, and in particular it has come to outperform Phrase-Based Machine Translation (PBMT) and PBMT-NMT combinations in the most recent WMT shared news translation tasks (Bojar et al., 2016, 2017). Taking this into account, we decided to build an NMT system taking as a basis the Transformer architecture, which has been shown to provide state-of-the-art SMT results while requiring relatively short times to train (Vaswani et al., 2017).

Apart from the SMT system itself, we also describe our work on parallel-corpus preprocessing and filtering, an aspect which has gained importance in WMT18 with the addition of the much larger and noisier parallel corpus ParaCrawl. Regarding data augmentation, we report as well how we extended the provided parallel dataset with data based on synthetic source sentences generated from the provided target-language monolingual corpora (in compliance with this shared task's "constrained" conditions).

This paper is organized as follows: in Section 2, we outline the data preparation techniques that were used (corpus preprocessing, corpus filtering, and data augmentation with synthetic source sentences); Section 3 shows the architecture and parameters of our NMT system and our system combination; in Section 4, we report our experiments and results (including on data preparation and on final system evaluation); and we draw our final conclusions in Section 5.

## 2 Data preparation

In this section, we describe the techniques that we used to prepare the provided WMT18 German↔English data (parallel and monolingual) to improve our SMT system results: corpus preprocessing (Section 2.1), corpus filtering (Section 2.2) and parallel data augmentation with synthetic source sentences (Section 2.3).

Corpus preprocessing and filtering has acquired a new relevance in WMT18, due to the addition of the new ParaCrawl parallel corpus, which sextuplicates the amount of German↔English parallel data that was available in WMT17 and previous editions: there are approx. 36 million sentence pairs in ParaCrawl, versus approx. 6 million in the rest of the parallel corpora (for a total sum of approx. 42 million sentence pairs in the full WMT18 training data). This is illustrated in Table 1, which summarizes the number of sentences of each corpus in the provided parallel dataset.

While the large size of the ParaCrawl parallel corpus makes it a valuable resource for system training in WMT18, it is much noisier than

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 418–424
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64041

Table 1: Size by corpus of the WMT18 parallel dataset

| Corpus | Sentences (M) |
|---|---|
| News Commentary v13 | 0.3 |
| Rapid (press releases) | 1.3 |
| Common Crawl | 1.9 |
| Europarl v7 | 2.4 |
| ParaCrawl | 36.4 |
| WMT18 total | 42.3 |

the rest of the WMT corpora. By noise here we mean misaligned sentences, wrong languages, meaningless sentences…; that is, sentence pairs which hinder system training for the purpose of German→English translation. In our experiments, we have observed that preprocessing and filtering the ParaCrawl corpus is necessary in order to make it useful as training data with the goal of increasing translation quality. In fact, using the ParaCrawl corpus "as is", we not only did not find any improvement in translation quality, but we even observed a degradation in all metrics of quality (as we will detail in Section 4.2).

Regarding data augmentation, the usage of relevant in-domain monolingual data has been shown to be important in order to improve NMT system results (Sennrich et al., 2016a). The provided WMT18 dataset contains large amounts of monolingual data which we can take advantage of to increase system accuracy. This fact led us to use these monolingual resources to generate additional synthetic data from target-language sentences.

### 2.1 Corpus preprocessing

Our preprocessing was done as suggested by the WMT18 organization (WMT18 organizers, 2018) using the provided scripts, with punctuation normalization, tokenization, cleaning and truecasing using standard Moses scripts.

Additionally, we removed from the training corpus any sentence that contained strange characters, defined as those lying outside the Latin UTF interval (u0000-u20AC) plus the euro sign (€). This allowed us to reduce the vocabulary size by eliminating unnecesary characters belonging to languages other than German or English that are not required for the translation of online news.

### 2.2 Corpus filtering

In regard to data filtering, we aimed to filter out noisy sentence pairs from the parallel corpora. To this end, we trained two separate 9-gram character-based language models (one for German, one for English) on the newstest2014 development set, based on which we computed the perplexity for each sentence in the full WMT18 dataset (including ParaCrawl), in a manner similar to the techniques described by Yasuda et al. (2008), Foster et al. (2010) and Axelrod et al. (2011). The software used was the SRI Language Modelling Toolkit (Stolcke et al., 2011).

The idea was that the lower the perplexity for a given sentence with respect to a reference news test corpus, the lower the odds of this sentence being noise (for the purpose of training a German→English SMT system). At the same time, this method could be considered to provide some degree of domain adaptation, since we score the sentences with respect to an in-domain reference corpus.

To produce the final score for each sentence pair, we combined the perplexity scores $(s, t)$ with the geometric mean $(\sqrt{s \cdot t})$. The geometric mean of two character-based perplexities can be interpreted as the character-based perplexity of the concatenation, assuming both sentences have the same number of characters. This is usually not the case exactly, but it is a good enough approximation. As the square root is a monotonic function, it does not alter the order of the scores.

We then ranked all the sentence pairs in the full WMT18 dataset according to their combined perplexity score, and selected subsets of different sizes, taking in each case the $n$ lowest-scored (less noisy) sentence pairs.

### 2.3 Synthetic source sentences

We augmented the WMT18 German↔English parallel training dataset (while keeping it under "constrained" conditions) with synthetic source sentences generated from the provided target-language monolingual corpora. To this end, we followed the approach outlined by Sennrich et al. (2016a).

In particular, we trained an English→German NMT system based on our best system configuration for German→English. Then, we used this system to generate our synthetic source sentences (German) from a subset of the WMT18 target-language monolingual corpora (English), which provided us with a significant amount of new sentence pairs to use as in-domain synthetic training data.

## 3 System description

We decided to build an NMT system based on the Transformer architecture (Vaswani et al., 2017). We opted for a pure NMT system due to the great advances this technology has made in the field of SMT over the last few years, which has led it to outperform systematically the more traditional PBMT systems and PBMT-NMT combinations, as introduced in Section 1. In particular, the Transformer architecture, based on self-attention mechanisms, can provide state-of-the-art SMT results while keeping training times relatively short. Regarding the software used, we used the Sockeye NMT framework (Hieber et al., 2017).

We based our systems on the less complex Transformer "base" configuration, which has significantly fewer parameters than the "big" configuration (65M parameters in the former case, 213M in the latter), and is thus much quicker to train (in exchange for a relatively small decrease in translation quality, in the case of the experiments described by Vaswani et al. (2017)). This was important in order to complete our experiments and the final training of our primary system in time for participation in this shared task. Thus, our models use 6 self-attentive layers both on the encoder and the decoder, with a model dimension of 512 units and a feed-forward dimension of 2048 units.

During training, we applied 0.1 dropout and 0.1 label smoothing, the Adam optimization scheme (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and learning rate annealing: we set an initial learning rate of 0.0001, and scaled this by a factor of 0.7 whenever the validation perplexity did not improve in 8 consecutive checkpoints (each checkpoint being equivalent to 2000 parameter updates). The system was trained with a word-based batch size of 3000, and a maximum sentence length of 75 tokens (subword units).

For our internal experiments, all systems were trained after applying 20K BPE operations (Sennrich et al., 2016b); but when building our final submissions, we increased this amount to 40K BPE operations (this will be detailed for each system in Section 4.4).

The final system consists of an ensemble of 4 independent training runs of our best model, based on a linear combination of the individual probabilities.

## 4 Experimental evaluation

In this section, we outline our experimental setup (Section 4.1); we report our experiments and results on corpus filtering (Section 4.2); we detail our setup for parallel data augmentation with synthetic source sentences (Section 4.3); and we discuss our final German→English NMT system evaluation and results (Section 4.4).

### 4.1 Experimental setup

For our experiments, we used newstest2015 as the development set and newstest2017 as the test set. We also report the results obtained with this year's newstest2018.

We evaluated our systems using the BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) measures, using mteval from the Moses SMT toolkit (Koehn et al., 2007) and tercom (Snover et al., 2008), respectively. All reported scores are according to the instructions on system output formatting provided by the WMT18 organization.

### 4.2 Results on corpus filtering

We show here the results obtained with the corpus filtering techniques explained in Section 2.2.

Table 2 summarizes the results in translation quality obtained with different subsets of the WMT18 parallel dataset. We can see that using the full WMT18 parallel dataset (42M sentence pairs), including the ParaCrawl corpus "as is", leads to a significant degradation in all metrics of quality compared to using the WMT18 dataset excluding ParaCrawl (6M sentence pairs; our baseline system for system evaluations in Section 4.4). Furthermore, we see that if we restrict ourselves to an excessively small training dataset (5M sentence pairs) using our filtering approach, there is also a degradation in quality with respect to using the unfiltered WMT18 dataset excluding ParaCrawl (6M).

We can also see (focusing on the test set results, newstest2017) that our filtering approach is effective at selecting useful training data from ParaCrawl, in the fact that the filtered datasets with sizes over the baseline's 6M sentence pairs provide significant improvements in quality (even if we limit ourselves to the small increase in size of the 7.5M subset). At the other extreme, in our experiments, going over 15M filtered sentence pairs meant setting the threshold for noise too low, as quality metrics degraded again.

Table 2: Results of 9-gram character-based language model data filtering, by number of selected sentences

| | newstest2015 | | newstest2017 | | newstest2018 | |
|---|---|---|---|---|---|---|
| Subset (no. of parallel sentences) | BLEU | TER | BLEU | TER | BLEU | TER |
| Full WMT18 parallel dataset (42M) | 20.6 | 71.1 | 21.3 | 70.2 | 26.2 | 64.2 |
| Baseline: WMT18 minus ParaCrawl (6M) | 31.1 | 55.4 | 32.0 | 54.8 | 39.1 | 46.3 |
| Filtered corpus (5M) | 30.3 | 56.3 | 31.4 | 55.5 | 38.7 | 46.5 |
| Filtered corpus (7.5M) | 32.8 | 54.0 | 33.7 | 56.5 | 41.5 | 44.5 |
| Filtered corpus (10M) | 33.0 | 53.7 | 34.5 | 52.9 | 42.2 | 43.7 |
| Filtered corpus (15M) | 33.4 | 53.2 | 34.3 | 52.7 | 42.2 | 43.6 |

As Table 2 shows, we obtained our best test set results with the 10M and 15M subsets. As results were very similar in both cases, we considered that any possible improvements in quality obtained from using the larger 15M subset were too small to justify using it instead of the 33% smaller 10M subset (which has a significantly faster convergence time for system training). Thus, the 10M subset is the filtered training corpus we took as a basis for the subsequent work described in Sections 4.3 and 4.4.

As a downside, this data filtering method based on independent language models for each side of a noisy parallel corpus has the caveat of not being able to detect sentence pairs where the source and the target are valid sentences, but not actually a translation of each other. To avoid this problem, it could be useful to combine into the filtering method the score provided by a simple, quick translation model (which should provide better scores for the sentence pairs which are correctly aligned translations). While we carried out some preliminary experiments on filtering with this approach, we did not obtain conclusive improvements in time for this shared task, so we left this for future work.

We also left for future work further corpus filtering experiments with other data selection approaches, such as using the cross-entropy difference (rather than just perplexity or cross entropy) to score each sentence pair (Moore and Lewis, 2010), or the dynamic data selection method described by van der Wees et al. (2017).

### 4.3 Synthetic source sentence setup

Here we detail how we augmented the WMT18 German↔English parallel training dataset, based on the technique introduced in Section 2.3.

We created an English→German NMT system using our best parameters for German→English (as described in Section 3), and trained it with the 10M-sentence filtered WMT18 parallel dataset that had shown the best performance for German→English (as described in Section 4.2). For reference, the resulting English→German NMT system obtained 27.4 BLEU on newstest2017. While improving this "inverse" system with further experiments could result in better synthetic training data (Sennrich et al., 2016a), we settled on this configuration (which obtains reasonable results with respect to the best WMT17 systems) in order to be in time for participation in this shared task.

Then, using this system, we translated into German a random sample of 20M English sentences from News Crawl 2017 (the most recent in-domain corpus among the provided WMT18 monolingual corpora). This provided us with 20M sentence pairs of German→English in-domain synthetic training data.

This augmented corpus was used for the final systems the results of which are discussed in the following Section 4.4.

### 4.4 Final system evaluation and results

We will now describe the most significant results obtained with the German→English NMT models we trained for WMT18 (based on the architecture and parameters outlined in Section 3). These results are shown in Table 3.

Our baseline model was trained excluding the ParaCrawl corpus from the training data, since using the full WMT18 corpus (with ParaCrawl) actually led to worse results (as we saw in Section 4.2). As mentioned in Section 3, this system was trained with 20K BPE operations (as is the case with the next system we will describe).

Our first step to improve these baseline results was filtering the full WMT18 corpus (including ParaCrawl), as explained in Section 4.2. In Ta-

Table 3: Results of German→English MT system evaluations

| | newstest2015 | | newstest2017 | | newstest2018 | |
|---|---|---|---|---|---|---|
| System | BLEU | TER | BLEU | TER | BLEU | TER |
| Baseline (WMT18 minus ParaCrawl, 6M pairs) | 31.1 | 55.4 | 32.0 | 54.8 | 39.1 | 46.3 |
| Filtered corpus (including ParaCrawl, 10M pairs) | 33.0 | 53.7 | 34.5 | 52.9 | 42.2 | 43.7 |
| + Synthetic data (2*10M+20M pairs), 40K BPE | 34.3 | 52.0 | 35.9 | 51.2 | 44.7 | 41.1 |
| Ensemble (x4) | 34.6 | 51.9 | 36.2 | 51.0 | 45.1 | 40.8 |

ble 3 we show the result obtained with a system trained on our best filtered corpus. As we saw in Section 4.2, the 10M filtered corpus provides an improvement of 2.5 BLEU and 1.7 TER in the test set over the baseline model. This shows how our data-filtering approach has allowed us to extract useful sentences from the noisy ParaCrawl corpus and improve our system performance.

For our final systems, we added 20M synthetic sentence pairs as described in Section 4.3, and we oversampled the previous 10M filtered bilingual training set by duplicating it, which gave us a final training set with a total of 40M sentence pairs[1]. We also increased the number of BPE operations from 20K to 40K. A single system trained with this configuration obtained 35.9 BLEU and 51.2 TER in the test set. This represents a significant improvement of 1.4 BLEU and 1.7 TER over the previous model, explained by a combination of the additional sentence pairs and the increase in vocabulary size.

As reference of the training times required, training a system with this final configuration took approx. 120 hours on a single-GPU system (Nvidia GeForce GTX 1080 Ti)[2].

Finally, our primary submission for WMT18 consists of an ensemble of 4 independent training runs with this final configuration, resulting in 36.2 BLEU and 51.0 TER in our test set, and 45.1 BLEU and 40.8 TER in newstest2018.

## 5 Conclusions

The MLLP group of the Universitat Politècnica de València has participated in the German→English WMT18 news translation shared task with an ensemble of neural machine translation models based on the Transformer architecture. Our models were trained using a filtered subset of the provided parallel training dataset, plus augmented parallel data based on synthetic source sentences generated from the provided monolingual corpora. Our primary submission was an ensemble of four independent training runs of our best model parameters.

Our results point to the usefulness of the Transformer NMT architecture to obtain highly competitive SMT results with a relatively low computational cost (which can contribute to "democratizing" access to state-of-the-art research in NMT to a higher number of research groups, even those with more modest computational equipment). We have also shown the importance of adequate corpus filtering to make the most of larger, noisier parallel corpora, employing a simple approach to filtering using character-based language models that has resulted in significant improvements in translation quality.

## Acknowledgments

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 761758 (X5gon); the Spanish government's TIN2015-68326-R (MINECO/FEDER) research project MORE, university collaboration grant programme 2017-2018, and faculty training scholarship FPU13/06241; the Generalitat Valenciana's predoctoral research scholarship ACIF/2017/055; as well as the Universitat Politècnica de València's PAID-01-17 R&D support programme.

[1] Oversampling the 10M original training set was a measure intended to keep in check the weight of the comparatively large 20M synthetic training data. We left for future work experimenting with different ratios of synthetic versus original data, such as 1:1 (Sennrich et al., 2016a; Fadaee et al., 2017), as additional comparison terms to determine the best performing configuration.

[2] While our systems were trained on single-GPU machines, multi-GPU system training with proportionally larger batch sizes (larger than the 3000 words per batch we used, as noted in Section 3) could deliver better translation quality results (Vaswani et al., 2017). We left this for future work.

## References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain Adaptation via Pseudo In-domain Data Selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Stroudsburg, Pennsylvania, USA.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany.

Marzie Fadaee, Aarianna Bisazza, and Christof Monz. 2017. Data Augmentation for Low-Resource Neural Machine Translation. *ArXiv e-prints (arXiv:1705.00440)*.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Cambridge, Massachussets, USA.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *arXiv e-prints (arXiv:1712.05690)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference for Learning Representations*, San Diego, California, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180, Stroudsburg, Pennsylvania, USA.

Robert C. Moore and William Lewis. 2010. Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Stroudsburg, Pennsylvania, USA.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. 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*, pages 223–231, Cambridge, Massachussets, USA.

Matthew Snover, Shuguang Wang, and Spyros Matsoukas. 2008. Translation Error Rate. http://www.cs.umd.edu/˜snover/tercom/. [Online; accessed 6-July-2018].

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at Sixteen: Update and Outlook. In *Proceedings of the 2011 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Waikoloa, Hawaii, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic Data Selection for Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark.

WMT18 organizers. 2018. WMT18 Shared Task: Machine Translation of News. http://www.statmt.org/wmt18/translation-task.html. [Online; accessed 24-July-2018].

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumita. 2008. Method of Selecting Training Data to Build a Compact and Efficient Translation Model. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, pages 655–660, Hyderabad, India.

# Microsoft's Submission to the WMT2018 News Translation Task: How I Learned to Stop Worrying and Love the Data

**Marcin Junczys-Dowmunt**
Microsoft
1 Microsoft Way
Redmond, WA 98121, USA

## Abstract

This paper describes the Microsoft submission to the WMT2018 news translation shared task. We participated in one language direction – English-German. Our system follows current best-practice and combines state-of-the-art models with new data filtering (dual conditional cross-entropy filtering) and sentence weighting methods. We trained fairly standard Transformer-big models with an updated version of Edinburgh's training scheme for WMT2017 and experimented with different filtering schemes for Paracrawl. According to automatic metrics (BLEU) we reached the highest score for this subtask with a nearly 2 BLEU point margin over the next strongest system. Based on human evaluation we ranked first among constrained systems. We believe this is mostly caused by our data filtering/weighting regime.

## 1 Introduction

This paper describes the Microsoft submission to the WMT2018 (Bojar et al., 2018) news translation shared task. We only participated in one language direction – English-German. Our system follows current best-practice and combines state-of-the-art models with new data filtering and weighting methods. According to automatic metrics (BLEU) we reached the highest score for this subtask with a nearly 2 BLEU point margin over the next strongest system. We believe this is mostly caused by our data filtering/weighting regime. Based on human evaluation we ranked first among constrained systems.

Our title references the fact that we built fairly standard models, updating existing baselines for WMT2017 to the new Transformer model (Vaswani et al., 2017), but spent more time on data cleaning and work with Paracrawl. As a side-effect we came up with a new parallel data filtering method which we call dual conditional cross-entropy filtering.

## 2 The Marian toolkit

For our experiments, we use Marian (Junczys-Dowmunt et al., 2018) an efficient Neural Machine Translation framework written in pure C++ with minimal dependencies. Microsoft Translator employees are contributing code to Marian. In the evolving eco-system of open-source NMT toolkits, Marian occupies its own niche best characterized by two aspects:

- It is written completely in C++11 and intentionally does not provide Python bindings; model code and meta-algorithms are meant to be implemented in efficient C++ code.
- It is self-contained with its own back end, which provides reverse-mode automatic differentiation based on dynamic graphs.

Marian is distributed under the MIT license and available from `https://marian-nmt.github.io` or the GitHub repository `https://github.com/marian-nmt/marian`.

## 3 NMT architectures

In Junczys-Dowmunt et al. (2018), we prepared a baseline setup for Marian which reproduces the highest scoring NMT system (Sennrich et al., 2017) in terms of BLEU during the WMT 2017 shared task on English-German news translation (Bojar et al., 2017). We further replaced the original RNN-based architecture with Transformer-style models from Vaswani et al. (2017) corresponding to their "base" and "big" architectures. In this section, we reuse the recipe and the proposed models as a set of strong baselines.

### 3.1 Deep transition RNN architecture

The model architecture in Sennrich et al. (2017) is a sequence-to-sequence model with single-layer RNNs in both, the encoder and decoder. The RNN

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 425–430
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64042

| System | 2016 | 2017 | 2018* |
|---|---|---|---|
| Deep RNN (x1) | 34.3 | 27.7 | - |
| +Ensemble (x4) | 35.3 | 28.2 | - |
| +R2L Reranking (x4) | 35.9 | 28.7 | - |
| Transformer-base (x1) | 35.6 | 28.8 | 43.2 |
| +Ensemble (x4) | 36.4 | 29.4 | 44.0 |
| +R2L Reranking (x4) | 36.8 | 29.5 | 44.4 |
| Transformer-big (x1) | 36.6 | 30.0 | 44.2 |
| +Ensemble (x4) | 37.2 | 30.5 | 45.2 |
| +R2L Reranking (x4) | 37.6 | 30.7 | 45.5 |

Table 1: BLEU results for our replication of the UEdin WMT17 system for the en-de news translation task. We reproduced most steps and replaced the deep RNN model with Transformer models. Asterisk * marks post-submission evaluation.

in the encoder is bi-directional. Depth is achieved by building stacked GRU-blocks resulting in very tall RNN cells for every recurrent step (deep transitions). The encoder consists of four GRU-blocks per cell, the decoder of eight GRU-blocks with an attention mechanism placed between the first and second block. As in Sennrich et al. (2017), embeddings size is 512, RNN state size is 1024. We use layer-normalization (Ba et al., 2016) and variational drop-out with $p = 0.1$ (Gal and Ghahramani, 2016) inside GRU-blocks and attention.

### 3.2 Transformer architectures

We very closely follow the architecture described in Vaswani et al. (2017) and their "base" and "big" models.

### 3.3 Training recipe

Modeled after the description from Sennrich et al. (2017), we reuse the example available at `https://github.com/marian-nmt/marian-examples` and perform the following steps:

- preprocessing of training data, tokenization, true-casing[1], vocabulary reduction to 36,000 joint BPE subword units (Sennrich et al., 2016) with a separate tool.[2]
- training of a shallow model for back-translation on parallel WMT17 data;
- translation of 10M German monolingual news sentences to English; concatenation of artificial training corpus with original data (times two) to produce new training data;
- training of four left-to-right (L2R) deep models (either RNN-based or Transformer-based);
- training of four additional deep models with right-to-left (R2L) orientation; [3]
- ensemble-decoding with four L2R models resulting in an n-best list of 12 hypotheses per input sentence;
- rescoring of n-best list with four R2L models, all model scores are weighted equally;
- evaluation on newstest-2016 (validation set) and newstest-2017 with sacreBLEU.[4]

At this stage we did not update to WMT2018 parallel or monolingual training data. This might put us at a slight disadvantage, but we could reuse models and back-translated data that was produced earlier.

We train the deep RNN models and Transformer-base models with synchronous Adam on 8 NVIDIA Titan X Pascal GPUs with 12GB RAM for 10 epochs each. The back-translation model is trained with asynchronous Adam on 8 GPUs. The transformer-big models are trained until convergence on four NVIDIA P40 GPUs with 24GB RAM. We do not specify a batch size as Marian adjusts the batch based on available memory to maximize speed and memory usage. This guarantees that a chosen memory budget will not be exceeded during training and uses maximal batch sizes.

All models use tied embeddings between source, target and output embeddings (Press and Wolf, 2017). Contrary to Sennrich et al. (2017) or Vaswani et al. (2017), we do not average checkpoints, but maintain a continuously updated exponentially averaged model over the entire training run. Following Vaswani et al. (2017), the learning rate is set to 0.0003 (0.0002 for Transformer-big) and decayed as the inverse square root of the number of updates after 16,000 updates. When training the Transformer model, a linearly growing learning rate is used during the first 16,000 iterations, starting with 0 until the base learning rate is reached.

Table 1 contains our results for WMT2017 training data with back-translation. We match re-

[1] Proprocessing was performed using scripts from Moses (Koehn et al., 2007).

[2] `https://github.com/rsennrich/subword-nmt`

[3] R2L training, scoring or decoding does not require data processing, right-to-left inversion is built into Marian.

[4] `https://github.com/mjpost/sacreBLEU`

sults from Sennrich et al. (2017) with our reimplementation of their models (Deep RNN) and outperform them with base and big Transformer versions. Differences between the best Deep RNN model and Transformer-big reach up to 2 BLEU points for the complete system. Ensembling is quite effective, right-to-left reranking seems to be moderately effective for Transformer models.

## 4 Taking advantage of Paracrawl

This year's shared task included a new, large but somewhat noisy parallel resource: Paracrawl. First experiments with shallow RNN models (chosen for fast experimentation) indicated that adding this data without a rigorous data filtering scheme would lead to catastrophic loss in quality (compare WMT+back-trans and Paracrawl-32M in Table 2). We therefore experiment with data selection and weighting.

### 4.1 Dual conditional cross-entropy filtering

The scoring method introduced in this section is our main contribution to the WMT2018 Shared Task on Parallel Corpus Filtering (Koehn et al., 2018), details are provided in our corresponding system submission (Junczys-Dowmunt, 2018).

For a sentence pair $(x, y)$ we calculate a score:

$$\begin{aligned} &|H_A(y|x) - H_B(x|y)| \\ +&\frac{1}{2}\left(H_A(y|x) + H_B(x|y)\right) \end{aligned} \tag{1}$$

where $A$ and $B$ are translation models trained on the same data but in inverse directions, and $H_M(\cdot|\cdot)$ is the word-normalized conditional cross-entropy of the probability distribution $P_M(\cdot|\cdot)$ for a model $M$:

$$\begin{aligned} H_M(y|x) &= -\frac{1}{|y|}\log P_M(y|x) \\ &= -\frac{1}{|y|}\sum_{t=1}^{|y|}\log P_M(y_t|y_{<t}, x). \end{aligned}$$

The score (denoted as dual conditional cross-entropy) has two components with different functions: the absolute difference $|H_A(y|x) - H_B(x|y)|$ measures the agreement between the two conditional probability distributions, assuming that (word-normalized) translation probabilities of sentence pairs in both directions should be roughly equal. We want disagreement to be low, hence this value should be close to 0.

However, a translation pair that is judged to be equally improbable by both models will also have a low disagreement score. Therefore we weight the agreement score by the average word-normalized cross-entropy from both models. Improbable sentence pairs will have higher average cross-entropy values.

This score is also quite similar to the dual learning training criterion from He et al. (2016) and Hassan et al. (2018). The dual learning criterion is formulated in terms of joint probabilities, later decomposed into translation model and language model probabilities. In practice, the influence of the language models is strongly scaled down which results in a form more similar to our score.

While Moore and Lewis filtering requires an in-domain data set and a non-domain-specific data set to create helper models, we require a clean, relative high-quality parallel corpus to train the two dual translation models. We sample 1M sentences from WMT parallel data excluding Paracrawl and train Nematus-style translation models $W_{\text{de}\to\text{en}}$ and $W_{\text{en}\to\text{de}}$.

Formula (1) produces only positive values with 0 being the best possible score. We turn it into a partial score with values between 0 and 1 (1 being best) by negating and exponentiating, setting $A = W_{\text{de}\to\text{en}}$ and $B = W_{\text{en}\to\text{de}}$:

$$\begin{aligned} \text{adq}(x, y) = \exp(-(&|H_A(y|x) - H_B(x|y)| \\ &+ \frac{1}{2}\left(H_A(y|x) + H_B(x|y)\right))). \end{aligned}$$

We score the entire Paracrawl data with this score and keep the scores. We further assign a value of 1 to all original WMT parallel sentences. That way we have a score for every sentence.

### 4.2 Cross-entropy difference filtering

We treated cross-entropy filtering proposed by Moore and Lewis (2010) as another score. Cross-entropy filtering or Moore-Lewis filtering uses the quantity

$$H_I(x) - H_N(x) \tag{2}$$

where $I$ is an in-domain model, $N$ is a non-domain-specific model and $H_M$ is the word-normalized cross-entropy of a probability distribution $P_M$ defined by a model $M$:

$$\begin{aligned} H_M(x) &= -\frac{1}{|x|}\log P_M(x) \\ &= -\frac{1}{|x|}\sum_{t=1}^{|x|}\log P_M(x_t|x_{<t}). \end{aligned}$$

Sentences scored with this method and selected when their score is below a chosen threshold are likely to be more in-domain according to model $I$ and less similar to data used to train $N$ than sentences above that threshold.

We chose WMT German news data from the years 2015-2017 as our in-domain, clean language model data and sampled 1M sentences to train model $I = W_{\text{en}}$. We sampled 1M sentences from Paracrawl without any previously applied filtering to produce $N = P_{\text{de}}$.

To create a partial score for which the best sentence pairs produce a 1 and the worst at 0, we apply a number of transformations. First, we negate and exponentiate cross-entropy difference arriving at a quotient of perplexities of the target sentence $y$ ($x$ is ignored):

$$\begin{aligned}\operatorname{dom}'(x, y) &= \exp(-(H_I(y) - H_N(y))) \\ &= \frac{\mathrm{PP}_N(y)}{\mathrm{PP}_I(y)}.\end{aligned}$$

This score has the nice intuitive interpretation of how many times sentence $y$ is less perplexing to the in-domain model $W_{\text{de}}$ than to the out-of-domain model $P_{\text{de}}$.

We further clip the maximum value of the score to 1 (the minimum value is already 0) as:

$$\operatorname{dom}(x, y) = \max(\operatorname{dom}'(x, y), 1). \qquad (3)$$

This seems counterintuitive at first, but is done to avoid that a high monolingual in-domain score strongly overrides bilingual adequacy; we are fine with low in-domain scores penalizing sentence pairs. This is a precision-recall trade-off for adequacy and we prefer precision.

We score the entire parallel data, Paracrawl, back-translated data and previous WMT data with this score. Next we multiply the adequacy and domain-based score to obtain a single score for all parallel data and all Paracrawl data in particular.

### 4.3 Data selection

Based on the scores produced in the previous section, we sort the new Paracrawl data by decreasing scores from 1 to 0. Next we select the first N sentences from the sorted corpus, add it to WMT and back-translated data and train again a shallow RNN model. In our experiments it seems, that selecting the first 8M out of 32M sentences according to this score leads to the largest gains on WMT2016 test data. A loss of 2.5 BLEU on full WMT+Paracrawl data is turned into a gain of 1.4 BLEU on WMT with selected Paracrawl data (see +Paracrawl-8M in Table 2).

| Data | 2016 |
|---|---|
| WMT+back-trans. | 32.6 |
| +Paracrawl-32M | 30.1 |
| +Paracrawl-2M | 33.2 |
| +Paracrawl-4M | 33.5 |
| **+Paracrawl-8M** | **34.0** |
| +Paracrawl-16M | 31.9 |
| +Paracrawl-24M | 30.3 |
| **+Paracrawl-8M-weights** | **34.2** |
| +Paracrawl-24M-weights | 33.4 |

Table 2: Effects of data cleaning, filtering and weighting on BLEU. Evaluated with default shallow Nematus-style RNN model

### 4.4 Data weighting

We further experiment with sentence instance weighting, a feature available in Marian. Here we use the computed score for a sentence pair as a multiplier of the per-sentence cross-entropy cost during training. Sentences with high scores will contribute more to the training, sentence with low cost contribute less. Scores are however clipped at 1, so no score can contribute more than it would without weighting. As stated above, sentences from original WMT training data and from back-translation have an adequacy score of 1, so they are only weighted by their domain multiplier. Sentences from Paracrawl are weighted by a product of their adequacy and domain score. We see slight improvements for +Paracrawl-8M-weights over the unweighted version. It also seems that weighting can at least partially eliminate harmful effects from bad data. The 24M variant is far less damaging than the unweighted version. This seems worth to be explored in future work.

## 5 Final submission

We chose the +Paracrawl-8M-weights setting as our training setting for the Transformer-big configuration. Training and model parameters remain the same, we only add 8M Paracrawl sentences and sentence-level scores for all parallel sentences and retrain all models. In Table 3, we see that compared to Table 1 the Transformer-big model can

| System | 2016 | 2017 | 2018* |
|---|---|---|---|
| Transformer-big (x1) | 38.6 | 31.3 | 46.5 |
| +Ensemble (x4) | 39.3 | 31.6 | 47.9 |
| +R2L Reranking (x4) | 39.3 | 31.7 | 48.0 |
| **+Transformer-LM** | **39.6** | **31.9** | **48.3** |

Table 3: Best model retrained on WMT and selected Paracrawl data. Sentences are weighted. Asterisk * marks post-submission evaluation.

take even more advantage of the filtered, selected and weighted data than the shallow models we used for development. We gain 1 to 2.5 BLEU points on the different test sets. Right-to-left re-ranking seems to matter less, however these models had not yet fully converge at time of submission.

### 5.1 Ensembling with a Transformer-style language model

We also experiment with shallow-fusion[5] or ensembling with a language model. We train a Transformer-style language model with Marian, following the architecture of the Transformer-big decoder without target-source attention blocks. We observed that this type of model has lower perplexity than LSMT models with similar numbers of parameters. We use 100M German monolingual sentences from 2016-2018 news data and train for two full epochs.

The resulting language model is ensembled with the left-to-right translation models at decoding time. We determine an optimal weight of 0.4 on a the newstest2016. The other models in the ensemble have a weight of 1. Since scores are summed it is a 4 to 0.4 ratio for translation models versus language model log probabilities. We see that the language model has a small, but consistently positive effect on all test sets of 0.2-0.3 BLEU.

## 6 Results

According to the automatically calculated BLEU scores on the WMT submission page, we achieve the highest BLEU score for English-German by a large margin over the next best system. We include the results for the 7 best systems in Table 4. The next best systems are quite tightly packed. We also rank highest among constrained systems based on human evaluation (Table 5).

| System | BLEU |
|---|---|
| **Microsoft-Marian** | **48.3** |
| UCAM | 46.6 |
| NTT | 46.5 |
| KIT | 46.3 |
| MMT-PRODUCTION | 46.2 |
| UEDIN | 44.4 |
| JHU | 43.4 |

Table 4: Automatic BLEU scores from submission page for 7 best submissions. There were 21 submissions in total.

| Rank | Ave. % | Ave. z | System |
|---|---|---|---|
| 2 | **81.9** | **0.551** | **Microsoft-Marian** |
| | 82.3 | 0.537 | UCAM |
| | 80.2 | 0.491 | NTT |
| | 79.3 | 0.454 | KIT |
| 8 | 76.7 | 0.377 | JHU |
| | 76.3 | 0.352 | UEDIN |
| 11 | 71.8 | 0.213 | LMU-NMT |
| 15 | 36.7 | -0.966 | RWTH-UNSUP |
| 16 | 32.6 | -1.122 | LMU-UNSUP |

Table 5: Human evaluation of constrained systems. Unconstrained systems have been omitted, see Bojar et al. (2018) for full list.

## 7 Conclusions

It seems strong state-of-the-art models and data hacking are winning combinations. Our data filtering method – developed first for this system – also proved very effective during the Parallel Corpora Filtering Task and we believe it had a large influence on our current result.

## References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. arXiv preprint arXiv:1607.06450.

Ondrej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, and Julia Kreutzer, editors. 2017. Proc. of the 2nd Conference on Machine Translation, WMT 2017. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck,

[5] We do not like this term, in the end this is just ensembling.

Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers, Brussels, Belgium. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In Advances in neural information processing systems, pages 1019–1027.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. CoRR, abs/1803.05567.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 820–828. Curran Associates, Inc.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers, Brussels, Belgium. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In Proceedings of ACL 2018, System Demonstrations, pages 116–121. Association for Computational Linguistics.

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. 2007. Moses: Open source toolkit for statistical machine translation. In ACL. The Association for Computer Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers, Brussels, Belgium. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In Proceedings of the ACL 2010 Conference Short Papers, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, volume 2, pages 157–163.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's neural MT systems for WMT17. In Proceedings of the Second Conference on Machine Translation, WMT 2017, pages 389–399.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc.

# CUNI Submissions in WMT18

**Tom Kocmi** **Roman Sudarikov** **Ondřej Bojar**

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
<surname>@ufal.mff.cuni.cz

## Abstract

We participated in the WMT 2018 shared news translation task in three language pairs: English-Estonian, English-Finnish, and English-Czech. Our main focus was the low-resource language pair of Estonian and English for which we utilized Finnish parallel data in a simple method. We first train a "parent model" for the high-resource language pair followed by adaptation on the related low-resource language pair. This approach brings a substantial performance boost over the baseline system trained only on Estonian-English parallel data. Our systems are based on the Transformer architecture. For the English to Czech translation, we have evaluated our last year models of hybrid phrase-based approach and neural machine translation mainly for comparison purposes.

## 1 Introduction

This paper describes the Charles University's submission to WMT 2018 Shared Task: Machine Translation of News.

We have experimented with three language pairs: Czech *(CS)*, Estonian *(ET)* and Finnish *(FI)* paired with English *(EN)*. Altogether, we covered five directions: both direction for English-Estonian, both directions for English-Finnish and English to Czech translation.

Our main focus is improving the low-resource language translation and therefore we concentrate on the English and Estonian language pair with the help of Finnish-English parallel data. The Finnish is a good candidate since it is closely related to the Estonian language but considerably more training data are available.

For the Finnish and English language pair, we use standard Neural Machine translation *(NMT)* system Transformer (Vaswani et al., 2017) with model averaging.

Our last language pair of interest is English to Czech translation, where we use our last year's model Sudarikov et al. (2017) for comparison purposes. The system is based on a hybrid combination of phrase-based, transfer-based and NMT approaches.

The structure of the paper is the following. In Section 2, we describe the setup of our main systems for Estonian and Finnish. Section 3 presents the English-Czech model. Section 4 is devoted to the description of our datasets. Section 5 details the results achieved by our systems. Section 6 discusses other works in the area of multi-lingual translation systems. And finally Section 7 concludes the paper.

## 2 Estonian and Finnish Setup

The main focus of our participation is improving low-resource language Estonian with the use of Finnish data. Our method consists of first training a "parent" high-resource model and continue the training on the "child" (low-resource) parallel data as a means of model adaptation.

### 2.1 Low-Resource Language Adaptation

We present a method that uses related high-resource language pair as a boost in performance for a low-resource language pair. The method needs relies on only one condition and that is a vocabulary shared across all the languages in the parent as well as child language pairs.

The shared vocabulary is obtained by combining all training data when the vocabulary is generated. To avoid bias in the vocabulary towards the high-resource language pair, we use only as many sentence pairs from the high-resource pair as are available for the low-resource pair, calling this approach "balanced vocabulary". We did not experiment with other proportions of data.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 431–437
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64043

Our method is based on transfer learning (also called "adaptation" or "finetuning"). It starts with training of the parent high-resource language pair (English-Finnish in our case) until it reaches its best performance or is trained for sufficiently long. Then, the training corpus is switched to the low-resource language pair (English-Estonian) for the rest of the training, without resetting any of the training hyperparameters. Note that we are not resetting even the state of the adaptive learning rate. As mentioned in Kocmi and Bojar (2018), if the learning rate is reset, this approach stops working.

As such, this method is very similar to the transfer learning proposed by Zoph et al. (2016) and improved by the using the shared vocabulary as in Nguyen and Chiang (2017). Moreover, in contrast to those two papers, we show that this simple style of transfer learning can be used on both sides (i.e. either the source or the target language), not only with the target language common to both parent and child model. More details of our method are described in Kocmi and Bojar (2018).

This method does not need any modification of existing NMT frameworks. The only requirement is to use the shared vocabulary across both language pairs (we use vocabulary of wordpieces, Johnson et al., 2017). This is achieved by learning the wordpiece segmentation from the concatenated source and target sides of both the parent and child language pair.

All other parameters of the model can stay the same as for the standard NMT training.

### 2.2 Model Description

We use the Transformer model (Vaswani et al., 2017) which translates through an encoder-decoder framework, with each layer involving an attention network followed by a feed-forward network. The architecture is much faster than other NMT due to the absence of recurrent and convolutional layers.

The Transformer model seems superior to other NMT approaches as documented in e.g. Popel and Bojar (2018) and also several language pairs in the manual evaluation of WMT18 (Bojar et al., 2018).[1]

We use the Transformer sequence-to-sequence model as implemented in Tensor2Tensor (Vaswani et al., 2018) version 1.4.2. Our models are based on the "big single GPU" configuration as defined in the paper. We set the batch size to 2300 and maximum sentence length to 100 wordpieces, in order to fit the model to our GPUs (NVIDIA GeForce GTX 1080 Ti with 11 GB RAM).

We use exponential learning rate decay with the starting learning rate of 0.2 and 32000 warm-up steps. Decoding uses the beam size of 8 and length normalization penalty is set to 1.

## 3 Chimera Description

For English-Czech translation task, we took the same system combination setup as described in Sudarikov et al. (2017). We used outputs of three different individual forward translation systems, trained on a synthetic backtranslated training dataset and combined them into the final output. These systems are Chimera2016 (Tamchyna et al., 2016; Bojar et al., 2016b), NeuralMonkey (Helcl et al., 2018)[2] and Marian (where the translation part was formerly known as AmuNMT) (Junczys-Dowmunt et al., 2016) with pretrained English-to-Czech Nematus models.[3] All the used datasets are described in Section 4.

The outputs of the two neural systems, consisting of translations of WMT15–18 test sets, were used to extract additional phrase tables for Moses. These tables were added to the Chimera2016 system, which already had one phrase table from genuine parallel data and one synthetic phrase table from TectoMT (Žabokrtský et al., 2008) output. After that, we used MERT (Och, 2003) to estimate the weights for Moses alternative decoding paths with multiple translation tables. MERT was run on the WMT16 test set. Further details on experiments with different combinations of phrase tables are available in Sudarikov et al. (2017).

## 4 Data Preparation

This section describes the data used for the training of our models. First, we describe training data for Estonian and Finnish.

There are many different sources for WMT18 News shared task that are allowed for the constrained task. We used most of the allowed data but decided to drop some sources.

For the Estonian-English, we use Europarl and Rapid corpora. We did not use Paracrawl because

[1] http://www.statmt.org/wmt18/translation-task.html

[2] http://ufal.mff.cuni.cz/neuralmonkey

[3] http://data.statmt.org/rsennrich/wmt16_systems

| Language pair | Sentences |
|---|---|
| Estonian-English | 0.8 M |
| Finnish-English | 2.8 M |
| Czech-English | 71.7 M |
| Estonian mono-news | 2.6 M |
| Finnish mono-news | 12.0 M |
| Czech mono-news | 59.2 M |

Table 1: Overview of training datasets. The top half lists sentence pair counts for parallel corpora and the bottom half the sentence counts of monolingual data.

we find it very noisy. The development set is from WMT News 2018.

The Finnish-English was prepared as in Östling et al. (2017), removing Wikipedia headlines. The dev set is from WMT News 2015.

We dropped sentence pairs shorter than 4 words or longer than 75 words on either source or target side to allow for a speedup of Transformer training by capping the maximal sentence length and increasing the batch size. Our experiments showed no translation performance change due to the reduction of the training data.

For English-Czech models, we used the same datasets as described in Sudarikov et al. (2017). First we took Czech monolingual news corpus, which was translated into English using Nematus (Sennrich et al., 2017) model, with 59 million sentences. We also used the genuine parallel data extracted from CzEng 1.6 (Bojar et al., 2016a) using the XenC toolkit (Rousseau, 2013) with Czech monolingual news corpus as the reference in-domain text. That part gave us additinal 12M sentences. The same monolingual news corpus was used for the language models.

The final data sizes are presented in Table 1.

### 4.1 Backtranslated Data

The organizers of WMT 2018 provide participants with vast amounts of monolingual data to use in translation systems, both in-domain and out-of-domain. We exploit the in-domain monolingual data for training as described by Sennrich et al. (2016) and previously suggested for PBMT e.g. by Bojar and Tamchyna (2011).

The idea is to translate the target side the monolingual data by an already trained machine translation system for the opposite translation direction and then use the synthetic data as a parallel corpus for the training of the main system. In this setup, the synthetic side is used as the input and the original monolingual sentences serve as the target.

Specifically, for the examined language pair EN→FI, we backtranslate monolingual Finnish data with the FI→EN model and mix the synthetic data with the available parallel EN→FI data to create the training corpus for EN→FI.

Sennrich et al. (2016) motivates the use of monolingual data with domain adaptation, due to the usage of in-domain monolingual data, reducing overfitting, and better modeling of fluency. Bojar and Tamchyna (2011) explain how backtranslation (with some fall-back for unknown words) allows to improve the vocabulary when targetting morphologically rich languages.

We get monolingual News Crawl data from all years of both Finnish and Estonian. We created the synthetic data from all monolingual data; we only drop sentences shorter than 6 words or longer than 75 words.

The monolingual data sizes are presented in Table 1.

It is important to stress that all the results in this paper are *without* the use of backtranslation. Only Table 4 presents the results with the use of backtranslated data.

## 5 Results and Discussion

In this section, we first present the results for Estonian-English and Finnish-English language pairs, focusing on transfer learning from the high-resource language pair to low-resource one. At the end, we compare the current NMT outputs to our last year's system for English to Czech translation.

The scores are evaluated by uncased SacreBLEU (Post, 2018).

We have computed statistical significance with pairwise bootstrap resampling with 1000 samples and alpha equal to 0.05 (Koehn, 2004).

Table 2 presents the effect of transfer learning from the parent model to the child model. The improvement is noticeable in both sides: the language unique to the child model can appear in the source or in the target.

Whenever the child language pair has more resources than the parent (Finnish-English in our case), the improvement is small or even (insignificantly) negative, as in ETEN-FIEN.

One could argue that the languages are too related and simply using the high-resource language pair model could work for the low-resource test sentences. The second column of Table 2 shows that this is not the case: the parent model without

| Parent - Child | Baseline | Only Parent | Transfer |
|---|---|---|---|
| ENFI - ENET | 17.03 | 2.32 | 19.74‡ |
| FIEN - ETEN | 21.74 | 2.44 | 24.18‡ |
| ENET - ENFI | 19.50 | 2.04 | 20.07‡ |
| ETEN - FIEN | 24.40 | 1.94 | 23.95 |
| ETEN - ENET | 17.03 | 1.41 | 17.46 |
| ENET - ETEN | 21.74 | 1.01 | 22.04‡ |

Table 2: Uncased BLEU scores for transfer learning of child models on various combinations of parent and child. The baseline is obtained by training only on the child parallel data. "Only Parent" represent result when no adaptation of parent model is done, i.e. running MT for the wrong language. The results are only comparable within each row. Results significantly better than the baseline are marked with ‡.

| Child Training Sents | Child BLEU | Baseline BLEU |
|---|---|---|
| 800k | 19.74 | 17.03 |
| 400k | 19.04 | 14.94 |
| 200k | 17.95 | 11.96 |
| 100k | 17.61 | 9.39 |
| 50k | 15.95 | 5.74 |
| 10k | 12.46 | 1.95 |

Table 3: The maximal score reached by the English-to-Estonian child models for decreasing sizes of child's training data, trained on an English to Finnish parent (all models build upon the same parent ENFI after 800k steps trained on the whole ENFI training set). The baselines use only the reduced English-Estonian data.

any transfer learning does not work for translation of the child test set.

With this result in mind, we also tested the effect of using only the low-resource language pair in both directions: first as a parent trained in the reverse direction, followed by training of the child on the same parallel corpus, now in the intended direction. The results of this can be seen in the bottom part of Table 2. It is an interesting result that only by using the low-resource data twice (in the reverse and then the correct direction), we could get a small boost in performance, significant when targetting ETEN.

In Table 3, we simulate extremely low-resource languages by downscaling the data for the child model. The smaller the child data, the bigger relative improvement is obtained. A reasonable performance is obtained even with as few as 10k sentence pairs in the child. This result suggests that when dealing with the very low-resource language, it is useful to utilize a related language pair as a pre-training parent step.

|  | Only | Transfer | With Backtranslated |  |
|---|---|---|---|---|
| Language Pair | Parallel | learning | Equal Size | All |
| EN-ET | 17.03 | **19.74** | **21.43** | **22.73‡** |
| EN-FI | 19.50 | - | 22.96 | 23.57‡ |

Table 4: Results with backtranslated data, either up to the size of the original parallel corpus ("Equal Size") or all available ("All"). The significance is computed between "Equal Size" and "All". The bold results are with additional use of transfer learning.

| Language pair | Baseline | Submitted |
|---|---|---|
| FI-EN | 21.52 | 21.52 |
| EN-FI | 15.13 | 15.13 |
| ET-EN | 20.68 | 23.50 |
| EN-ET | 16.54 | 19.49 |

Table 5: WMT18 newstest BLEU scores for the baseline runs and the runs submitted as "CUNI-Kocmi-*" for manual evaluation.

### 5.1 Effect of Backtranslation

The size of the training set can be extended also with the backtranslated data. We experiment with backtranslation only for two language directions: English to Estonian and English to Finnish.

First, we trained FI→EN and ET→EN models on parallel data for each of the language pairs. With those models, we translated all monolingual data. Finally, we mixed the synthetic and genuine parallel corpora for FI→EN and (separately) for ET→EN.

Table 4 presents our experiment with two setups. We either used only a subset of the synthetic corpus of the size equal to the genuine parallel data, or we use all available synthetic data. The former approach results in a training corpus with half of monolingual backtranslated data and half of original parallel texts. The latter approach results in parallel training set containing 76.5% monolingual data for Estonian and 81.1% for Finnish. In both cases, we report the score on the dev set after 600k steps of training.

The motivation for applying this upper bound is that the synthetic corpus could introduce more translation errors and damage translation quality. The results in Table 4 however document that this is not the case and more data is better.

### 5.2 Estonian and Finnish Submitted Models

Our submitted models for Finnish and Estonian are presented in Table 5, with the baseline of no transfer. Unfortunately, we submitted models without backtranslation for manual evaluation.

| Language pair | WMT17 | WMT18 |
|---|---|---|
| CUNI-Transformer | 23.8 | 26.0 |
| UEDIN-NMT | 22.8 | 23.4 |
| CUNI-Chimera2017 | 20.5 | 19.8 |

Table 6: Cased-BLEU results from `matrix.statmt.org`.

For Finnish, the submitted models did not include the transfer learning step so the FI→EN and EN→FI Baseline and Submitted scores are identical.

The Estonian-to-English model was trained from the Finnish-to-English model at its 800k training steps. The English-to-Estonian built upon the English-to-Finnish, trained also for 800k steps.

### 5.3 English-to-Czech Benchmark

Table 6 shows cased-BLEU scores for WMT17 and WMT18 test sets as presented at `http://matrix.statmt.org`.[4]

The Chimera setup remains the same in both years, so it can serve as a reference point, documenting the improvement of other systems. The gap between Chimera and the best neural systems considerably widened in terms of BLEU score (from +2.3 on WMT17 to +3.6 on WMT18 when comparing to UEDIN-NMT and from +3.3 to +6.2 when comparing to CUNI-Transformer).

## 6 Related Work

Firat et al. (2016) propose zero-resource multi-way multilingual systems, with the main goal of reducing the total number of parameters needed to train multiple source and target languages. To keep all the language pairs "active" in the model, a special training schedule is needed. Otherwise, catastrophic forgetting would remove the ability to translate between the languages trained earlier.

Johnson et al. (2017) test another multilingual approach: all translation pairs are simply used at once and the desired target language is indicated with a special token at the end of the source side. The model implicitly learns translation between many languages and it can even translate among language pairs never seen together.

The lack of parallel data can be tackled by unsupervised translation (Artetxe et al., 2018; Lample et al., 2018). The general idea is to mix monolingual training of autoencoders for the source and target languages with translation trained on data translated by the previous iteration of the system.

Aside from the common back-translation (Sennrich et al., 2016), simple copying of target monolingual data back to source (Currey et al., 2017) has been also shown to improve translation quality in low-data conditions.

Similar to transfer learning is also curriculum learning (Bengio et al., 2009; Kocmi and Bojar, 2017), where the training data are ordered from foreign out-of-domain to the in-domain training examples.

## 7 Conclusion

In this paper, we presented our systems for WMT 2018 shared news translation task in three language pairs: English-Estonian, English-Finnish, and English-Czech.

English-Estonian was the main focus of our research, with the English-Finnish used to improve the quality of the translations. Both Finnish and Estonian systems used the Transformer architecture. Our results show that a simple transfer learning is beneficial. Further gains (not in of our submitted systems) were obtained by including back-translated data.

Our English-Czech submission was prepared and used mainly for comparison purposes and it showed the widening gap between hybrid phrase-based and neural systems.

## Acknowledgments

This study was supported in parts by the grants SVV 260 453, GAUK 8502/2016, and 18-24210S of the Czech Science Foundation. This work has been using language resources and tools stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (projects LM2015071 and OP VVV VI CZ.02.1.01/0.0/0.0/16 013/0001781).

## References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural ma-

[4] `http://matrix.statmt.org/matrix/systems_list/1867` for 2017 and `http://matrix.statmt.org/matrix/systems_list/1883` for 2018.

chine translation. In *Proceedings of the Sixth International Conference on Learning Representations*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016a. Czeng 1.6: enlarged czech-english parallel corpus with processing tools dockered. In *International Conference on Text, Speech, and Dialogue*, pages 231–238. Springer.

Ondrej Bojar, Roman Sudarikov, Tom Kocmi, Jindrich Helcl, and Ondrej Cıfka. 2016b. Ufal submissions to the iwslt 2016 mt track. *IWSLT. Seattle, WA*.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving Translation Model by Monolingual Data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, Scotland. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Jindřich Helcl, Jindřich Libovickỳ, Tom Kocmi, Tomáš Musil, Ondřej Cífka, Dusan Varis, and Ondřej Bojar. 2018. Neural monkey: The current state and beyond. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, volume 1, pages 168–176.

Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernand a Vigas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions. In *Proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*, Seattle, WA.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum Learning and Minibatch Bucketing in Neural Machine Translation. In *Recent Advances in Natural Language Processing 2017*.

Tom Kocmi and Ondřej Bojar. 2018. Trivial Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 3rd Conference on Machine Translation (WMT)*, Brussels, Belgium.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, volume 4, pages 388–395.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301. Asian Federation of Natural Language Processing.

Franz Josef Och. 2003. 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.

Robert Östling, Yves Scherrer, Jörg Tiedemann, Gongbo Tang, and Tommi Nieminen. 2017. The helsinki neural machine translation system. In *Proceedings of the Second Conference on Machine Translation*, pages 338–347, Copenhagen, Denmark. Association for Computational Linguistics.

Martin Popel and Ondej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. *arXiv preprint arXiv:1804.08771*.

Anthony Rousseau. 2013. Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, 100:73–82.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a Toolkit for Neural Machine

Translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Roman Sudarikov, David Mareček, Tom Kocmi, Dušan Variš, and Ondřej Bojar. 2017. CUNI Submission in WMT17: Chimera Goes Neural. In *Proceedings of the 2nd Conference on Machine Translation (WMT)*, Copenhagen, Denmark.

Aleš Tamchyna, Roman Sudarikov, Ondřej Bojar, and Alexander Fraser. 2016. CUNI-LMU submissions in WMT2016: Chimera constrained and beaten. In *Proceedings of the First Conference on Machine Translation, Berlin, Germany. Association for Computational Linguistics*.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for Neural Machine Translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 193–199, Boston, MA. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. TectoMT: Highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

# The JHU Machine Translation Systems for WMT 2018

Anonymous EMNLP submission

## Abstract

We report on the efforts of the Johns Hopkins University to develop neural machine translation systems for the shared task for news translation organized around the Conference for Machine Translation (WMT) 2018. We developed systems for German–English, English–German, and Russian–English. Our novel contributions are iterative back-translation and fine-tuning on test sets from prior years.

## 1 Introduction

We carried out two relatively independent efforts on German–English language directions and Russian–English, using the Marian and Sockeye neural machine translation toolkits, respectively.

The German–English systems outperformed last year's best result (37.0 vs. 35.1 (+1.9) for German–English, 29.1 vs. 28.3 (+0.8) for English–German), but fell short against this year's best performing systems (45.3 vs. 48.4 (-3.1) and 43.4 vs. 48.3 (-4.9), respectively)[1]. The best models this year used the Transformer model instead of the recurrent neural networks that our models are based on. Our novel contributions are iterative back-translation and fine-tuning on prior test sets.

For Russian–English, we carried out extensive hyperparameter search, with different numbers of layers, embedding and hidden state sizes, and drop-out settings.

## 2 German–English and English–German

The systems for the German–English language pairs were developed with the Marian toolkit (Junczys-Dowmunt et al., 2018). We developed models with both shallow and deep architectures, based on recurrent neural networks. We ensembled 4 independent runs and reranked with right-to-left models (output in reverse order). We saw improvements with iterative back-translations and fine tuning on test sets from previous years, as well as use of the Paracrawl corpus (unfiltered).

A big challenge for system development are long training times (a month on a single GTX 1080ti GPU) which limited our ability to exploit the Paracrawl corpus. Because of this, we also started system development almost a year ago, using the training data from last year for the most part. All scores reported in this Section are on `newstest2017` with case-sensitive BLEU.

### 2.1 Shallow System Development

We started with shallow systems similar to Edinburgh's submission two years ago (Sennrich et al., 2016a). It uses byte pair encoding with a vocabulary of 50,000 (Sennrich et al., 2016c) and back-translation of the `news2016` monolingual corpus (Sennrich et al., 2016b), about twice the size of the original training data.

For each training run, we compare different ways to obtain a single best model.

- Use the single model that performed best on the dev set (newstest2016).
- Use checkpoint ensembling to obtain the 4 or 8 best models, and decode the test set with an ensemble of these models.
- Merge the models obtained by checkpoint ensembling into a single model.

For German–English, we achieved slightly better results with an ensemble of independent models rather than a merged model (about +0.2 BLEU), while for English–German they perform similarly. Ensembling of either kind clearly outperforms the single best model.

We then built ensembles of the resulting systems for the 4 independent runs. This gives gains

[1] Scores reported at `http://matrix.statmt.org/`

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 438–444
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64044

**Shallow German–English**

| | single | ensemble | | merged | |
|---|---|---|---|---|---|
| | | 4 | 8 | 4 | 8 |
| run 1 | 31.8 | 32.3 | 32.5 | 32.2 | 32.3 |
| run 2 | 32.4 | 32.6 | 32.8 | 32.5 | 32.7 |
| run 3 | 32.8 | 32.8 | 32.7 | 32.5 | 32.5 |
| run 4 | 32.2 | 32.9 | 32.9 | 32.7 | 32.7 |
| ensemble | 33.2 | | | 33.2 | 33.3 |
| r2l rerank | | | | 33.7 | |

**Deep German–English**

| | single | ensemble | | merged | |
|---|---|---|---|---|---|
| | | 4 | 8 | 4 | 8 |
| run 1 | 34.5 | 34.6 | 34.4 | 35.1 | 35.1 |
| run 2 | 34.2 | 34.3 | 34.2 | 34.3 | 34.3 |
| run 3 | 34.5 | 34.3 | 34.3 | 34.5 | 34.5 |
| run 4 | 34.0 | 34.3 | 34.3 | 34.9 | 34.6 |
| ensemble | 34.9 | | | 35.6 | 35.6 |
| r2l rerank | | | | 35.7 | |

**Shallow English–German**

| | single | ensemble | | merged | |
|---|---|---|---|---|---|
| | | 4 | 8 | 4 | 8 |
| run 1 | 25.9 | 26.2 | 26.2 | 26.1 | 26.1 |
| run 2 | 25.5 | 26.1 | 26.1 | 25.9 | 25.9 |
| run 3 | 25.6 | 25.6 | 25.7 | 25.6 | 25.7 |
| run 4 | 25.3 | 25.7 | 25.8 | 25.8 | 25.8 |
| ensemble | 26.4 | | | 26.4 | 26.5 |
| r2l rerank | | | | 27.3 | |

**Deep English–German**

| | single | ensemble | | merged | |
|---|---|---|---|---|---|
| | | 4 | 8 | 4 | 8 |
| run 1 | 27.7 | 28.0 | 28.1 | 27.8 | 27.9 |
| run 2 | 27.7 | 27.8 | 27.7 | 27.8 | 27.7 |
| run 3 | | 28.0 | 27.8 | 27.9 | 27.8 |
| run 4 | 26.1 | 27.5 | 27.8 | 27.8 | 27.9 |
| ensemble | 28.3 | | | 28.3 | 28.3 |
| r2l rerank | | | | 28.9 | |

Table 1: Shallow and deep systems for German-English with Marian. 4 independent training runs, with checkpoint ensemble, and merging the checkpoint ensemble into a single model (averaging parameters). Ensemble of the runs, with right-to-left reranking (4 independent right-to-left runs).

of about +0.5 over the merged checkpoint ensembles. Notable, the ensemble over the single systems yields essentially the same quality.

The final improvement is right-to-left reranking (Liu et al., 2016) where we built also 4 independent systems on the data sets with the output word order reversed. This gave improvements of +0.5 for German–English and +0.9 for English–German. For detailed results, see Table 1.

### 2.2 Deep System Development

System development for deep models is essentially the same as for shallow models. We used the same data sets, also carried out 4 independent runs for each language direction, carried out checkpoint ensembling for each run, combined the resulting models in a ensemble and performed reranking with right-to-left models.

The models are similar to Edinburgh's submission from last year (Sennrich et al., 2017). They use 4 alternating encoder layers and 4 decoder layers, LSTM cells, dropout, layer normalization, tied embeddings, and Adam optimization.

Detailed results are also in Table 1. Merging the checkpoint models worked better for German–English, and about the same for English–German, compared to decoding with the multiple models. Ensembling the 4 independent runs yielded solid gains (about +0.5), but reranking helped substantially only for English–German (+0.6).

### 2.3 Iterative Backtranslation

The back-translated data was generated with a single shallow system trained on the parallel data. Since we obtained much better performance by using this back-translated data, employed deep model architecture and ensembled independent runs, we have now a much better system to back-translate data.

Note that this second round of backtranslation uses monolingual data in both languages. Starting with a German–English system (trained on parallel data), we translate monolingual German news text. We then use this synthetic parallel corpus to build a English–German system (in addition to the provided parallel data). We now use this English–German system to translate monolingual English text, yielding again a synthetic parallel corpus to be used in the final system.

We carried out the same system development

**Iterative Deep German–English**

| | single | ensemble | | merged | |
|---|---|---|---|---|---|
| | | 4 | 8 | 4 | 8 |
| run 1 | 35.5 | 35.6 | 35.6 | 35.6 | 35.6 |
| run 2 | 35.3 | 35.6 | 35.6 | 35.7 | 35.6 |
| run 3 | 35.6 | 35.6 | 35.5 | 35.6 | 35.7 |
| run 4 | 35.1 | 35.5 | 35.4 | 35.6 | 35.7 |
| ensemble | 36.1 | | | 36.1 | 36.1 |
| r2l rerank | | | | 36.5 | |

**Iterative Deep English–German**

| | single | ensemble | | merged | |
|---|---|---|---|---|---|
| | | 4 | 8 | 4 | 8 |
| run 1 | 28.5 | 28.5 | 28.5 | 28.5 | 28.5 |
| run 2 | 28.1 | 28.2 | 28.3 | 28.3 | 28.3 |
| run 3 | 27.8 | 28.1 | 28.3 | 28.3 | 28.4 |
| run 4 | 28.6 | 28.5 | 28.6 | 28.7 | 28.4 |
| ensemble | 29.1 | | | 29.0 | 28.9 |
| r2l rerank | | | | 29.4 | |

Table 2: System development with deep models using iterative back-translation.

as for the shallow and deep models. See Table 2 for details. Table 3 shows how the quality of the back-translation impacts the final system's performance. This table is also reported in our paper on iterative back-translation (Hoang et al., 2018).

## 2.4 Use of Paracrawl Corpus

For German–English only, we added the Paracrawl corpus without any filtering to the training data used up to this point (parallel data plus iterative back-translated monolingual data). We only completed one training run, and obtained 36.3 BLEU (ensembled 4 checkpoints) as opposed to 35.6–35.7 for models without Paracrawl.

We trained this model for more than 2 months on a single GTX1080ti GPU. The best dev score (newstest2016) after 2 weeks was 43.0, after 4 weeks 43.3, after 6 weeks 43.7, and after 10 weeks 43.8. So, it seems to be necessary to train such a model for at least a month and a half.

Adding this model to the ensemble gives a score of 36.6. Weighting the Paracrawl model as much as all the 4 models without gives slightly higher score than equal weights (36.5 for equal weights).

**German–English**

| | back | final |
|---|---|---|
| no back-translation | - | 29.6 |
| 10k iterations | 10.6 | 29.6 (+0.0) |
| 100k iterations | 21.0 | 31.1 (+1.5) |
| convergence | 23.7 | 32.5 (+2.9) |
| re-back-translation | 27.9 | 33.6 (+4.0) |
| + deep ensemble | | 36.1 (+6.2) |

**English–German**

| | back | final |
|---|---|---|
| no back-translation | - | 23.7 |
| 10k iterations | 14.5 | 23.7 (+0.0) |
| 100k iterations | 26.2 | 25.2 (+1.5) |
| convergence | 29.1 | 25.9 (+2.2) |
| re-back-translation | 34.8 | 27.0 (+3.3) |
| + deep ensemble | | 29.0 (+5.3) |

Table 3: Impact of the quality of the back-translation system on the final system performace. Note that the back-translation systems run in the opposite direction and are not comparable to the numbers in the same row. The *deep ensemble* scores reported here match results in Table 2.

**German–English**

| Setup | BLEU |
|---|---|
| iterative back runs 1–4 | 35.6–35.7 |
| run with Paracrawl | 36.3 |
| ensemble | 36.6 |
| + fine-tuned | 37.0 |

Table 4: Final refinements: a model trained with the unfiltered Paracrawl corpus, an ensemble of the 4 iterative back-translation models, plus the

## 2.5 Fine Tuning on Prior Test Sets

Finally, we fine tuned the model of one of the four iterative backtranslation runs towards the test sets from previous years. We trained for 3 epochs with a learning rate of 0.0003. Adding the resulting model to the ensemble gives an additional gain of +0.4, resulting in a final score of 37.0.

# 3 Russian–English

## 3.1 Data

We use the provided bitext for training neural machine translation systems (NMT) in a constrained setting. The bitext is first pre-processed via Joshua's[2] normalize.pl, followed by tokenize.pl and lowercase.pl. The training data is additionally filtered to sentences less than 80 tokens, result-

[2] http://joshua.incubator.apache.org/

ing in 37M sentence pairs (777M English tokens, 725M Russian tokens). We use newstest2016 (2998 sentence pairs) as the development set for early-stopping during NMT training. For continued training experiments, we further used a concatenation of newstests from 2012 to 2016 (14822 sentence pairs). We did not exploit any additional monolingual data, either by itself or via back-translation. After tokenization and lowercasing, the training data consists of 4.6M Russian and 3.7M English vocabulary types. We ran BPE[3]independently for each language, with 50K merge operations each.

All results in this section are reported on newstest2017 (3001 sentence pairs), which is treated as the initial test set. Unless otherwise specified, we report BLEU scores from multi-bleu.perl directly computed on lower-cased tokenized English reference.

### 3.2 Setup

In this task, we use Sockeye version 1.18.1[4] (Hieber et al., 2018) as our NMT engine. We explored a three-step approach to model building:

1. Hyperparameter search: First, we trained multiple NMT models using different hyperparameter settings (e.g. #layers, embedding size) on the 37M-sentence training bitext.

2. Continued training: Second, we attempted to improve the independent models in Step 1 via continued training on the newstest2012-2016 data, which more closely matches the test set in terms of domain.

3. Ensembles: Finally, we took the best models in Step 2 and performed ensemble decoding.

All our NMT systems above are sequence-to-sequence models using LSTM units. For training, we use the ADAM optimizer, with training set perplexity as the objective. The initial learning rate is set to 0.0003, and reduces by a factor of 0.5 after 3 checkpoints without improvement of development perplexity (”plateau-reduce” scheduler). The checkpoint is computed at a frequency of every 10k batches; with a batch size of 128 sentences, this corresponds to 1280k sentences, or 1/29th of the training data. After 8 checkpoints without improvements, the training is deemed to have converged. Most training runs converged between 30 to 100 checkpoints, which corresponds to 1 to 3 epochs over the training data. We then use the checkpoint with the best validation perplexity as the chosen model for each run. For decoding, we use beam search with the default beam size of 5.

### 3.3 Hyperparameter Search

We searched over four types of hyperparameters:

- The number of stacked LSTM layers in the encoder and decoder: **layer**=$\{1, 2, 3\}$
- The dimension of the word embeddings in source and target: **embed**=$\{500, 1000\}$
- The number of hidden units in each LSTM: **embed**=$\{500, 1000\}$
- The dropout rate for the embedding layer: **dropout**=$\{0.1, 0.3\}$

The goal is to quantify how sensitive the results are to hyperparameter settings, and to find the best model for submission. We train systems for a sample of 9 different hyperparameter settings from the $3 \times 2 \times 2 \times 2 = 24$ total combinations, and summarize their results in Table 5. For convenience of exposition, we label these models with id a-i.

**Observation 1:** We observe there is a *large variance* of test-bleu scores among models a-i, ranging from 31.1 for model a (best) to 27.3 for model i (worst). This suggests that hyperparameter search is very important for building strongly performing systems, even for settings that are not too different.

For example, compare the smallest model (e), which has 80M trainable weights, to the second smallest model (a), which has 85M trainable weights: the only difference between the two is one extra layer and 5M extra weights, yet the test-bleu changes from 29.5 (e) to 31.1 (a). Similarly, compare model c (137M weights) to model g (141M weights): they differ only in one extra layer, yet test-bleu varies as much as 30.1 (c) to 27.9 (g). The largest model (f), which has 200M trainable weights, ranks in the middle in terms of test-bleu among the 9 models.

While it may be tempting to extract ”suggested hyperparameter settings” from Table 5, we recommend a more robust strategy is to perform hyperparameter search to the extent possible.

**Observation 2:** We find that perplexity correlates well with bleu when ranking models in hyperparameter search. To a large extent, models a-d, which have the best training perplexities

[3] https://github.com/rsennrich/subword-nmt/

[4] https://github.com/awslabs/sockeye

| | Hyperparameter Setting | | | | Results | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **id** | **layer** | **embed** | **hidden** | **dropout** | **test-bleu** | **step** | **train-ppl** | **dev-ppl** | **dev-bleu** |
| a | 2 | 500 | 500 | .1 | 31.1 | 91 | 5.20 | 9.13 | 27.7 |
| b | 2 | 1000 | 500 | .3 | 30.3 | 61 | 5.53 | 9.48 | 26.8 |
| c | 2 | 1000 | 500 | .1 | 30.1 | 58 | 5.37 | 9.39 | 27.3 |
| d | 1 | 1000 | 1000 | .3 | 29.9 | 57 | 5.37 | 8.98 | 27.2 |
| e | 1 | 500 | 500 | .1 | 29.5 | 72 | 5.68 | 10.36 | 26.2 |
| f | 3 | 1000 | 1000 | .1 | 28.2 | 28 | 6.37 | 10.35 | 25.8 |
| g | 3 | 1000 | 500 | .1 | 27.9 | 32 | 5.99 | 11.15 | 25.2 |
| h | 1 | 1000 | 500 | .1 | 27.4 | 36 | 6.28 | 11.92 | 24.7 |
| i | 1 | 1000 | 500 | .3 | 27.3 | 34 | 6.67 | 12.25 | 24.4 |
| a’ | 2 | 500 | 500 | .1 | 29.1 | 110 | 5.26 | 8.86 | 27.1 |
| e’ | 1 | 500 | 500 | .1 | 28.0 | 109 | 5.78 | 10.1 | 25.3 |

Table 5: Hyperparameter search results. The model with **id**=a is a sequence-to-sequence model with 2 **layer** of LSTMs in both the encoder and decoder, 500-dimensional source and target word embeddings (**embed**), 500 **hidden** units in each of the LSTM, and 0.1 **dropout** rate at the embedding layer. This model achieved 31.1 BLEU (**test-bleu**) on the test set (`newstest2017`) and comes from 91th **step** (or checkpoint) of the training run, which achieved a training set perplexity (**train-ppl**) of 5.20, a development set perplexity (**dev-ppl**) of 9.13, and a development set BLEU score (**dev-bleu**) of 27.7. All models with id a-j are trained on the BPE bitext with 50k merge operations in Russian and 50k merge operations in English, and are ranked in this table in terms of test-bleu. The last two rows represent additional experiments with model a’ and e’, which is similar to model a and e but are trained on BPE bitext with *30k* merge operations in Russian and 50k merge operations in English.

(**train-ppl**) and development perplexities (**dev-ppl**), also achieve the best BLEU scores (**dev-bleu**, **test-bleu**). This suggests that for hyperparameter search purposes, optimizing and validating based on perplexity is a sufficiently good surrogate for BLEU, which is expensive to compute.

**Observation 3:** The last two rows of Table 5 experiments with a different number of BPE operations for the source side (Russian). Models a’ and e’ are similar to models a and e, except that they use 30k merge operations rather than the 50k we used in all other experiments. The goal is to test the impact of subword units in hyperparameter search. The **train-ppl** and **dev-ppl** of these 30k models are better than or on-par with the 50k counterparts, but the BLEU scores appear to be worse. It is somewhat difficult to conclude with only these two datapoints, but we think that perhaps hyperparameter search with different subword units need to be conducted separately. Even on the same dataset, hyperparameters that work well in one version of the BPE data may not necessarily work well in another version of BPE.

**Observation 4:** It appears that the better models (a-e) seem to have trained longer; their final checkpoints are chosen at a relative high number of steps. For example, model (a) comes from checkpoint 91, which corresponds to 3 epochs over a 37M sentence dataset. In Figure 1, we plot the development BLEU for each of the training runs over time. Our models train for a maximum of 5 days; this is when the learning rate becomes minuscule and the training process determines convergence. We observe that BLEU continuously improves (while at a slower pace), even towards the end of the training process. This suggests that it might be possible to extract further BLEU gains by adjusting the learning rate and convergence criteria, encouraging the training to continue longer.

### 3.4 Continued Training

The training bitext comes from multiple domains, while the focus of the test set is news. One may treat this problem as domain adaptation. Here, we experiment with continued training[5] (Luong and Manning, 2015). The idea is:

**Phase 1**: Train a model until convergence on the multi-domain training bitext, as done in Sec 3.3.

**Phase 2**: Use the model weights from Phase 1

[5] Also called fine-tuning by some works.

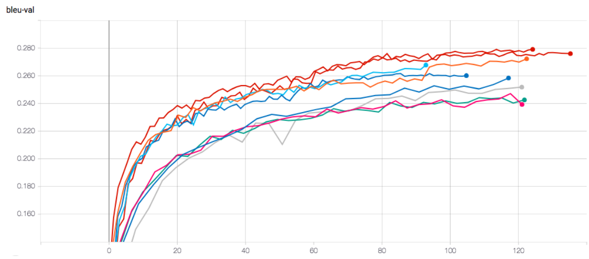


Figure 1: The 9 curves represent the change in BLEU scores when training each of models in Table 5. (y-axis: **dev-bleu**, x-axis: time in hours) Note that BLEU improves rapidly in the first 20 hours. The rate of improvement slows down but the improvement does not stop: BLEU continually improves even at hour 100 (4 days of training).

| id | base | 3 epochs | Δ | 9 epochs | Δ |
|---|---|---|---|---|---|
| a | 31.1 | 31.7 | 0.6 | 28.4 | -2.7 |
| b | 30.3 | 31.5 | 1.2 | 27.8 | -2.5 |
| c | 30.1 | 31.3 | 1.2 | 27.3 | -2.8 |
| d | 29.9 | 31.3 | 1.4 | 27.8 | -2.1 |
| e | 29.5 | 30.3 | 0.8 | 27.8 | -1.7 |
| g | 27.9 | 29.2 | 1.3 | 25.9 | -2.0 |
| h | 27.4 | 29.2 | 1.8 | 25.4 | -2.0 |
| i | 27.3 | 29.2 | 1.9 | 26.1 | -1.2 |

Table 6: Continued Training **test-bleu** on `newstest2017`. Base is the baseline number for each model from Table 5. The BLEU scores for continued training after 3 or 9 epochs are shown, along with their difference Δ against base. Continued training with few epochs improve results.

to initialize a new training process on adaptation data. This new training process usually only proceeds for a few steps. This is the Continued Training model, and can be used to decode the test set.

In Phase 2, we use `newstest2012-2016` as the adaptation training data. Part of it overlaps with the dev data (`newstest2016`), so the training procedure may constantly improve both train-ppl and dev-ppl, and never decide to converge. We therefore impose a hard-stop to prevent overfitting.

First, we experimented with stopping continued training after 3 epochs over the adaptation data. This corresponds to 350 batch updates (batch size is 128 sentences). ADAM is used as the optimizer, and the learning rate is fixed at a constant 0.0003.

The **test-bleu** scores are shown in Table 6. We observe that continued training is very effective, improving BLEU scores for all models by 0.6 to 1.9 points. For example, model (a) improves from 31.1 to 31.7, and model (b) improves from 30.3 to 31.5 on the `newstest2017` test set. However, if we train on adaptation data for too long, the results degrade. When continued training runs for 9 epochs (1150 batch updates), model (a) degrades to 28.4 The degradation is consistent for all models. This suggests that learning rate and amount of batch updates are important hyperparameters.

| ensemble | test-bleu | Δ |
|---|---|---|
| a+b+c+d+e+f (6) | 33.63 | 1.93 |
| a+b+c+d+e (5) | 33.57 | 1.87 |
| a+b+c+d (4) | 33.13 | 1.43 |
| a+b+c (3) | 33.13 | 1.43 |
| a+b (2) | 32.89 | 1.19 |

Table 7: Ensemble decoding **test-bleu** on `newstest2017`. We use the models from Table 6. The difference Δ is gain with respect to the best single model (a), with BLEU 31.7.

### 3.5 Ensembles

Finally, we performed ensemble decoding with the best continued training models obtained in the previous step. Table 7 shows a 6-model ensemble improves 1.93 BLEU over the best single model (a). This reaffirms the effectiveness of ensembles.

### 3.6 Final Russian–English Results

We submitted the 6-model ensemble in Table 7 as our final system in the official evaluation. As shown before, this model achieved 33.63 BLEU via multi-beu.perl on a tokenized and lowercased version of `newstest2017`. We also computed the official NIST-BLEU with the detokenized versions: it achieves 0.3195 (cased) and 0.3309 (lowercased) on `newstest2017`.

This result is only slightly improves upon our 2017 Moses SMT submission (Ding et al., 2017), which achieves 0.3129 (cased) and 0.3246 (lowercased) NIST-BLEU. We were interested in exploring the effectiveness of NMT under constrained data conditions (e.g. without backtranslation on large monolingual data) and standard sequence-to-sequence setups (e.g. withough reranking with left-to-right features or SMT/NMT hybrids). We imagine that these enhancements are needed if further gains are to be desired; unfortunately we may need to pay the cost of forgoing the simplicity of standard sequence-to-sequence NMT models.

## References

Shuoyang Ding, Huda Khayrallah, Philipp Koehn, Matt Post, Gaurav Kumar, and Kevin Duh. 2017. The jhu machine translation systems for wmt 2017. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 276–282, Copenhagen, Denmark. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2018. The sockeye neural machine translation toolkit at amta 2018. In *Annual Meeting of the Association for Machine Translation in the Americas (AMTA)*.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416, San Diego, California. Association for Computational Linguistics.

Minh-Thang Luong and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 76–79.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural MT systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

# JUCBNMT at WMT2018 News Translation Task: Character Based Neural Machine Translation of Finnish to English

**Sainik Kumar Mahata, Dipankar Das, Sivaji Bandyopadhyay**
Computer Science and Engineering
Jadavpur University, Kolkata, India
sainik.mahata@gmail.com,
dipankar.dipnil2005@gmail.com, sivaji_cse_ju@yahoo.com

## Abstract

In the current work, we present a description of the system submitted to WMT 2018 News Translation Shared task. The system was created to translate news text from Finnish to English. The system used a Character Based Neural Machine Translation model to accomplish the given task. The current paper documents the preprocessing steps, the description of the submitted system and the results produced using the same. Our system garnered a BLEU score of 12.9.

## 1 Introduction

Machine Translation (MT) is automated translation of one natural language to another using computer software. Translation is a tough task, not only for computers, but humans as well as it incorporates a thorough understanding of the syntax and semantics of both languages. For any MT system to return good translations, it needs good quality and sufficient amount of parallel corpus (Mahata et al., 2016, 2017).

In the modern context, MT systems can be categorized into Statistical Machine Translation (SMT) and Neural Machine Translation (NMT). SMT has had its share in making MT very popular among the masses. It includes creating statistical models, whose input parameters are derived from the analysis of bilingual text corpora, created by professional translators (Weaver, 1955). The state-of-art for SMT is Moses Toolkit[1], created by Koehn et al. (2007), incorporates subcomponents like Language Model generation, Word Alignment and Phrase Table generation. Various works have been done in SMT (Lopez, 2008; Koehn, 2009) and it has shown good results for many language pairs.

On the other hand NMT (Bahdanau et al., 2014), though relatively new, has shown considerable improvements in the translation results when compared to SMT (Mahata et al., 2018). This includes better fluency of the output and better handling of the Out-of-Vocabulary problem. Unlike SMT, it doesn't depend on alignment and phrasal unit translations (Kalchbrenner and Blunsom, 2013). On the contrary, it uses an Encoder-Decoder approach incorporating Recurrent Neural Cells (Cho et al., 2014). As a result, when given sufficient amount of training data, it gives much more accurate results when compared to SMT (Doherty et al., 2010; Vaswani et al., 2013; Liu et al., 2014).

Further, NMT can be of two types, namely Word Level NMT and Character Level NMT. Word Level NMT, though very successful, suffers from a few disadvantages. It are unable to model rare words (Lee et al., 2016). Also, since it does not learn the morphological structure of a language it suffers when accommodating morphologically rich languages (Ling et al., 2015). We can address this issue, by training the models with huge parallel corpus, but, this in turn, produces very complex and resource consuming models that aren't feasible enough.

To combat this, we plan to use Character level NMT, so that it can learn the morphological aspects of a language and construct a word, character by character, and hence tackle the rare word occurrence problem to some extent.

In the current work, we participated in the WMT 2018 News Translation Shared Task[2] that focused on translating news text, for European language pairs. The Character Based NMT system discussed in this paper was designed to accommodate Finnish to English translations. The orga-

[1] http://www.statmt.org/moses/

[2] http://www.statmt.org/wmt18/translation-task.html

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 445–448
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64045

nizers provided the required parallel corpora, consisting of 3,255,303 sentence pairs, for training the translation model. The statistics of the parallel corpus is depicted in Table 1 Our model was trained on a Tesla K40 GPU, and the training took around 10 days to complete.

| | |
|---|---|
| **# sentences in Fi corpus** | 3,255,303 |
| **# sentences in En corpus** | 3,255,303 |
| **# words in Fi corpus** | 53,753,718 |
| **# words in En corpus** | 73,694,350 |
| **# word vocab size for Fi corpus** | 1,065,309 |
| **# word vocab size for En corpus** | 280,822 |
| **# chars in Fi corpus** | 427,187,612 |
| **# chars in En corpus** | 405,624,094 |
| **# char vocab size for Fi corpus** | 963 |
| **# char vocab size for En corpus** | 1,360 |

Table 1: Statistics of the Finnish-English parallel corpus provided by the organizers. ”#” depicts No. of. ”Fi” and ”En” depict Finnish and English, respectively. ”char” means character and ”vocab” means vocabulary of unique tokens.

The remainder of the paper is organized as follows. Section 2 will describe the methodology of creating the character based NMT model and will include the preprocessing steps, a brief summary of the encoder-decoder approach and the architecture of our system. This will be followed by the results and conclusion in Section 3 and 4, respectively.

## 2 Methodology

For designing the model we followed some standard preprocessing steps, which are discussed below.

### 2.1 Preprocessing

The following steps were applied to preprocess and clean the data before using it for training our character based neural machine translation model. We used the NLTK toolkit[3] for performing the steps.

- **Tokenization**: Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens. In our case, these tokens were words, punctuation marks, numbers. NLTK supports tokenization of Finnish as well as English texts.
- **Truecasing**: This refers to the process of restoring case information to badly-cased or non-cased text (Lita et al., 2003). Truecasing helps in reducing data sparsity.
- **Cleaning**: Long sentences (# of tokens > 80) were removed.

### 2.2 Neural Machine Translation

Neural machine translation (NMT) is an approach to machine translation that uses neural networks to predict the likelihood of a sequence of words. The main functionality of NMT is based on the sequence to sequence (seq2seq) architecture, which is described in Section 2.2.1.

#### 2.2.1 Sequence to Sequence Model

Sequence to Sequence learning is a concept in neural networks, that helps it to learn sequences. Essentially, it takes as input a sequence of tokens (characters in our case)

$$X = \{x_1, x_2, ..., x_\mathrm{n}\}$$

and tries to generate the target sequence as output

$$Y = \{y_1, y_2, ..., y_\mathrm{m}\}$$

where $\mathrm{x_i}$ and $\mathrm{y_i}$ are the input and target symbols respectively.

Sequence to Sequence architecture consists of two parts, an Encoder and a Decoder.

The encoder takes a variable length sequence as input and encodes it into a fixed length vector, which is supposed to summarize its meaning and taking into account its context as well. A Long Short Term Memory (LSTM) cell was used to achieve this. The uni-directional encoder reads the characters of the Finnish texts, as a sequence from one end to the other (left to right in our case),

$$\vec{h}_\mathrm{t} = \vec{f}_\mathrm{enc}(E_\mathrm{x}(x_\mathrm{t}), \vec{h}_\mathrm{t\text{-}1})$$

Here, $\mathrm{E_x}$ is the input embedding lookup table (dictionary), $\vec{f}_\mathrm{enc}$ is the transfer function for the Long Short Term Memory (LSTM) recurrent unit. The cell state *h* and context vector *C* is constructed and is passed on to the decoder.

The decoder takes as input, the context vector *C* and the cell state *h* from the encoder, and computes the hidden state at time t as,

$$s_\mathrm{t} = f_\mathrm{dec}(E_\mathrm{y}(y_\mathrm{t\text{-}1}), s_\mathrm{t\text{-}1}, c_\mathrm{t})$$

[3] https://www.nltk.org/

Subsequently, a parametric function $\mathrm{out}_{\mathrm{k}}$ returns the conditional probability using the next target symbol $k$.

$$(y_{\mathrm{t}} = k \mid y < t, X) = \frac{1}{Z} exp(out_{\mathrm{k}}(E_{\mathrm{y}}(y_{\mathrm{t}}{-}1), s_{\mathrm{t}}, c_{\mathrm{t}}))$$

$Z$ is the normalizing constant,

$$\sum{}_{\mathrm{j}} exp(out_{\mathrm{j}}(E_{\mathrm{y}}(y_{\mathrm{t}} - 1), s_{\mathrm{t}}, c_{\mathrm{t}}))$$

The entire model can be trained end-to-end by minimizing the log likelihood which is defined as

$$L = -\frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{Ty^{\mathrm{n}}} logp(y_{\mathrm{t}} = y_{\mathrm{t}}{}^{\mathrm{n}}, y_{\mathrm{i}\mathrm{t}}{}^{\mathrm{n}}, X^{\mathrm{n}})$$

where N is the number of sentence pairs, and $\mathrm{X}^{\mathrm{n}}$ and $\mathrm{y}_{\mathrm{t}}{}^{\mathrm{n}}$ are the input sentence and the t-th target symbol in the n-th pair respectively.

The input to the decoder was one hot tensor (embeddings at character level) of English sentences while the target data was identical, but with an offset of one time-step ahead.

### 2.3 Training

For training the model, we preprocessed the Finnish and English texts to normalize the data. Thereafter, Finnish and English characters were encoded as One-Hot vectors. The Finnish characters were considered as the input to the encoder and subsequent English characters was given as input to the decoder. A single LSTM layer was used to encode the Finnish characters. The output of the encoder was discarded and only the cell states were saved for passing on to the decoder. The cell states of the encoder and the English characters were given as input to the decoder. Lastly, a Dense layer was used to map the output of the decoder to the English characters, that were mapped with an offset of 1. The *batch size* was set to 128, *number of epochs* was set to 100, activation function was *softmax*, optimizer chosen was *rmsprop* and loss function used was *categorical cross-entropy*. Learning rate was set to 0.001. The architecture of the constructed model is shown in Figure 1.

## 3 Results

Our system was a constrained system, which means that we only used data given by the organizers to train our system. The output was converted to an SGML format, the code for which was provided by the organizers. The results were submitted to http://matrix.statmt.org/ for evaluation. The organizers calculated the BLEU score, BLEU-cased score, TER score, BEER 2.0 score, and Character TER score for our submission. As for the human ranking scores, the system fetched a standardized Average $Z$ score of -0.404 and a non-standardized Average $\%$ score of 58.9 (Bojar et al., 2018). The results of the automated and human evaluation scores are given in Table 2.

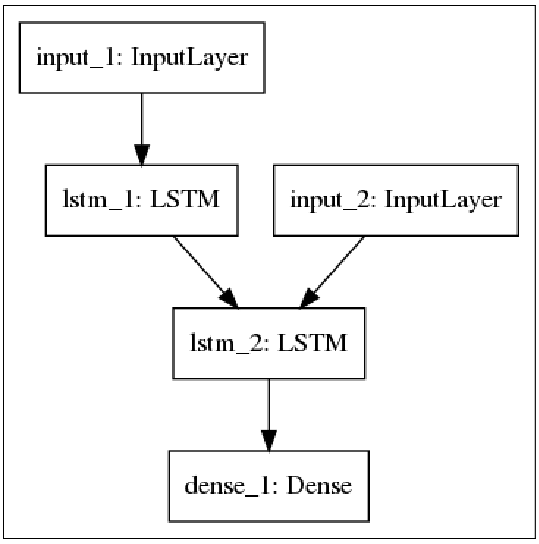


Figure 1: Architecture of the reported NMT model.

| Metrics | Score |
|---|---|
| **BLEU** | 12.9 |
| **BLEU Cased** | 12.2 |
| **TER** | 0.816 |
| **BEER 2.0** | 0.448 |
| **Character TER** | 0.770 |
| **Average** $Z$ | -0.404 |
| **Average** $\%$ | 58.9 |

Table 2: Evaluation Metrics

## 4 Conclusion

The paper presents the working of the translation system submitted to WMT 2018 News Translation shared task. We have used character based encoding for our proposed NMT system. We have used a single LSTM layer as an encoder as well as a decoder. As a future prospect, we plan to use more LSTM layers in our model. We plan to create another NMT model, which takes as input words, and not characters and subsequently use various embedding schemes to improve the translation quality.

## Acknowledgments

The reported work is supported by Media Lab Asia, MeitY, Government of India, under the Visvesvaraya PhD Scheme for Electronics & IT.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Stephen Doherty, Sharon O?Brien, and Michael Carl. 2010. Eye tracking as an mt evaluation technique. *Machine translation*, 24(1):1–13.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, volume 3, page 413.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *CoRR*, abs/1610.03017.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W. Black. 2015. Character-based neural machine translation. *CoRR*, abs/1511.04586.

Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. Truecasing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 152–159. Association for Computational Linguistics.

Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. 2014. A recursive recurrent neural network for statistical machine translation.

Adam Lopez. 2008. Statistical machine translation. *ACM Computing Surveys (CSUR)*, 40(3):8.

Sainik Mahata, Dipankar Das, and Santanu Pal. 2016. Wmt2016: A hybrid approach to bilingual document alignment. In *WMT*, pages 724–727.

Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay. 2017. Bucc2017: A hybrid approach for identifying parallel sentences in comparable corpora. *ACL 2017*, page 56.

Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay. 2018. Mtil2017: Machine translation using recurrent neural network on statistical machine translation. *Journal of Intelligent Systems*, pages 1–7.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *EMNLP*, pages 1387–1392.

Warren Weaver. 1955. Translation. *Machine translation of languages*, 14:15–23.

# NICT's Neural and Statistical Machine Translation Systems for the WMT18 News Translation Task

**Benjamin Marie** **Rui Wang**
**Atsushi Fujita** **Masao Utiyama** **Eiichiro Sumita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Souraku-gun, Kyoto, 619-0289, Japan
{bmarie, wangrui, atsushi.fujita, mutiyama, eiichiro.sumita}@nict.go.jp

## Abstract

This paper presents the NICT's participation to the WMT18 shared news translation task. We participated in the eight translation directions of four language pairs: Estonian-English, Finnish-English, Turkish-English and Chinese-English. For each translation direction, we prepared state-of-the-art statistical (SMT) and neural (NMT) machine translation systems. Our NMT systems were trained with the transformer architecture using the provided parallel data enlarged with a large quantity of back-translated monolingual data that we generated with a new incremental training framework. Our primary submissions to the task are the result of a simple combination of our SMT and NMT systems. Our systems are ranked first for the Estonian-English and Finnish-English language pairs (constraint) according to BLEU-cased.

## 1 Introduction

This paper describes the neural (NMT) and statistical machine translation systems (SMT) built for the participation of the National Institute of Information and Communications Technology (NICT) to the WMT18 shared News Translation Task (Bojar et al., 2018). We participated in four language pairs (eight translation directions): Estonian-English (Et-En), Finnish-English (Fi-En), Turkish-English (Tr-En), and Chinese-English (Zh-En). We chose these language pairs since they appear to be among the most challenging: involving distant languages and with less training data, for Finnish, Estonian, and Turkish, provided by the organizers than for Russian, German, and Czech. All our systems are *constrained*, i.e., we used only the parallel and monolingual data provided by the organizers to train and tune them. For all the translation directions, we trained NMT and SMT systems, and combined them through $n$-best list reranking using different informative features as proposed by Marie and Fujita (2018). This simple combination method, associated to the exploitation of large back-translated monolingual data, performed among the best MT systems at WMT18. Especially for the competitive Et-En and Fi-En translation tasks, for which our submissions are ranked first according to the BLEU-cased metric (henceforth BLEU). Our systems for Et-En, Fi-En, and Tr-En were trained using the exactly same procedures, without any specific linguistic treatments. On the other hand, for Zh-En, we used a specific tokenizer and used slightly different training parameters due to the much larger quantity of training data.

The remainder of this paper is organized as follows. In Section 2, we introduce the data preprocessing. In Section 3, we describe the details of our NMT and SMT systems. The back-translation of monolingual data using our new incremental training framework for NMT is described in Section 4. Then, the combination of NMT and SMT is described in Section 5. Empirical results produced with our systems are showed and analyzed in Section 6, and Section 7 concludes this paper.

## 2 Data Preprocessing

### 2.1 Data

As parallel data to train our systems, we used all the available data for all our targeted translation directions, except the "Wiki Headlines"[1] corpus for Fi-En. As English monolingual data, we used all the available data except the "Common Crawl" and "News Discussions" corpora.[2] For all other languages, we used all the available monolingual corpora, except for Turkish for which we

[1] It contains only very short segments that are not sentences and that we therefore assume to be of no use in NMT.

[2] The "News Crawl" data are sufficiently large and that these corpora are not in-domain monolingual data.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 449–455
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64046

| Language pair | #sent. pairs | #tokens | |
|---|---|---|---|
| Et-En | 1.9M | 29.4M (Et) | 36.0M (En) |
| Fi-En | 3.1M | 52.9M (Fi) | 72.8M (En) |
| Tr-En | 207.4k | 4.4M (Tr) | 5.1M (En) |
| Zh-En | 24.8M | 509.9M (Zh) | 576.2M (En) |

Table 1: Statistics of our preprocessed parallel data.

| Language | #lines | #tokens |
|---|---|---|
| En | 338.7M | 7.5B |
| Et | 146.1M | 3.6B |
| Fi | 177.1M | 3.2B |
| Tr | 105.0M | 1.8B |
| Zh | 130.5M | 2.3B |

Table 2: Statistics of our preprocessed monolingual data.

used only 100 millions sentence pairs randomly extracted from "Common Crawl."

To tune/validate and evaluate our systems, we used Newstest2016 and Newstest2017 for Fi-En and Tr-En, Newsdev2017 and Newstest2017 for Zh-En, and Newsdev2018 for Et-En.

### 2.2 Tokenization, Truecasing and Cleaning

We used Moses tokenizer (Koehn et al., 2007) and truecaser for English, Estonian, Finnish, and Turkish. The truecaser was trained on one million tokenized lines extracted randomly from the monolingual data. Truecasing was then performed on all the tokenized data. For Chinese, we used Jieba[3] for tokenization but did not perform truecasing. For cleaning, we only applied the Moses script `clean-n-corpus.perl` to remove lines in the parallel data containing more than 80 tokens and replaced characters forbidden by Moses. Note that we did not perform any punctuation normalization. Tables 1 and 2 present the statistics of the parallel and monolingual data, respectively, after preprocessing.

## 3 MT Systems

### 3.1 NMT

To build competitive NMT systems, we chose to rely on the transformer architecture (Vaswani et al., 2017) since it has been shown to outperform, in quality and efficiency, the two other mainstream architectures for NMT known as deep recurrent neural network (deep RNN) and convolutional neural network (CNN). We chose Marian[4] (Junczys-Dowmunt et al., 2018) to train and evaluate our NMT systems since it supports state-of-the-art features and is one of the fastest NMT framework publicly available.[5] In order to limit the size of the vocabulary of the NMT models, we segmented tokens in the parallel data into subword units via byte pair encoding (BPE) (Sennrich et al., 2016b) using 50k operations. BPE segmentations were jointly learned on the training parallel data for source and target languages, except for Zh-En for which Chinese and English segmentations were trained separately. All our NMT systems for Et-En, Fi-En, and Tr-En were consistently trained on 4 GPUs,[6] with the following parameters for Marian: `--type transformer --max-length 80 --mini-batch-fit --valid-freq 5000 --save-freq 5000 --workspace 8000 --disp-freq 500 --beam-size 12 --normalize 1 --valid-mini-batch 16 --overwrite --early-stopping 5 --cost-type ce-mean-words --valid-metrics ce-mean-words perplexity translation --keep-best --enc-depth 6 --dec-depth 6 --transformer-dropout 0.1 --learn-rate 0.0003 --dropout-src 0.1 --dropout-trg 0.1 --lr-warmup 16000 --lr-decay-inv-sqrt 16000 --lr-report --label-smoothing 0.1 --devices 0 1 2 3 --dim-vocabs 50000 50000 --optimizer-params 0.9 0.98 1e-09 --clip-norm 5 --sync-sgd --tied-embeddings --exponential-smoothing`. For Zh-En, we did not use `--dropout-src 0.1 --dropout-trg 0.1` since the training data is much larger. We performed NMT decoding with an ensemble of a total of six models according to the best BLEU (Papineni et al., 2002) and the best perplexity scores,[7] produced by three independent training runs.

[3] https://github.com/fxsjy/jieba

[4] https://marian-nmt.github.io/, version 1.4.0

[5] It is fully implemented in pure C++ and supports multi-GPU training.

[6] NVIDIA® Tesla® P100 16Gb.

[7] Note that the same model may give the best BLEU score and also the best perplexity score. Nonetheless, for consistency across language pairs, we systematically kept two models even if they were identical.

### 3.2 SMT

We also trained SMT systems using Moses. Word alignments and phrase tables were trained on the tokenized parallel data using mgiza. Source-to-target and target-to-source word alignments were symmetrized with the grow-diag-final-and heuristic. We trained hierarchical SMT models for Et-En and Fi-En since they provided better results than regular phrase-based models on our development data for these language pairs.[8] We also expected a similar observation for Tr-En and Zh-En. However, we were unable to exploit hierarchical models for the language pair Tr-En[9] while hierarchical models for the language pairs Zh-En were extremely large due to the size of our training data. Consequently, for Tr-En and Zh-En we simply trained regular phrase-based models using MSLR (monotone, swap, discontinuous-left, discontinuous-right) lexicalized reordering models and used the default distortion limit of 6. We trained two 4-gram language models: one on the entire monolingual data concatenated to the target side of the parallel data, and another one on the in-domain "News Crawl" corpora only, using LMPLZ (Heafield et al., 2013). For English, all singletons were pruned due to the large size of the monolingual data. To tune the SMT model weights, we used KB-MIRA (Cherry and Foster, 2012) and selected the weights giving the best BLEU score on the development data after 15 decoding runs.

## 4 Back-translation of Monolingual Data

### 4.1 Incremental Back-Translation with Et-En, Fi-En, and Tr-En

We introduced an incremental training framework for NMT aiming to iteratively increase the quality and quantity of the synthetic parallel data used for training. In this framework, we first simultaneously but independently train a source-to-target and a target-to-source NMT systems using the same original parallel data. Then, we back-translate source and target monolingual data respectively using the source-to-target and the target-to-source NMT systems, and obtain two sets of synthetic parallel data. And then, a new source-to-target and a new target-to-source NMT systems are trained, from scratch, on their respective new training data comprising the mixture of the original parallel data and the synthetic parallel data whose source side is back-translated from the target side. At this stage, we just do what is usually done by previous work (Sennrich et al., 2016a).

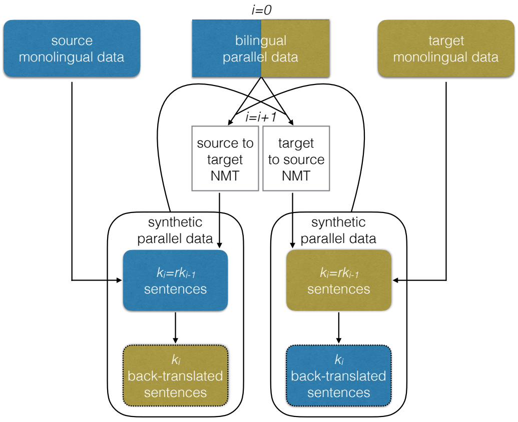


Figure 1: Our incremental training framework.

As illustrated in Figure 1, we continue this procedure iteratively. Using source-to-target and target-to-source NMT systems trained on the mixture of the synthetic and original parallel data, we back-translate a larger number of monolingual sentences, including the same sentences back-translated at the first iteration. Since we have better NMT systems than those at the first iteration, we can expect the back-translation to be of a better quality. We mix this new synthetic parallel data to the original one and train again from scratch a source-to-target and a target-to-source NMT systems to obtain further improved translation models. Note that this procedure is partially similar to the work proposed by Zhang et al. (2018) and Hoang et al. (2018), but differs in the sense that we increase incrementally our back-translated data.

Given the number of sentences used in the first iteration, $k_1$, and an expansion factor, $r$, we determine $k_i$, the number of monolingual sentences back-translated at iteration $i$, as follows:

$$k_i = rk_{i-1} \tag{1}$$

The parameters used for the given language pairs are listed in Table 3. The monolingual sentences to be back-translated were randomly extracted from the NewsCrawl corpora. For Et-En and Fi-En, we stopped the incremental training after 2 iterations, back-translating up to 2M sentences. For Tr-En, we observed improvements for

[8] Between 0.5 and 1 BLEU points of improvement.

[9] Moses consistently crashed (*segmentation fault*) during the decoding of the development data.

| Language pair | $k_1$ | $r$ | #iter. (total) |
|---|---|---|---|
| Et-En | 1M | 2 | 2 |
| Fi-En | 1M | 2 | 2 |
| Tr-En | 200k | 2 | 4 |

Table 3: Parameters used for our incremental training. For each language pair, the same parameters were used for both translation directions. In our preliminary experiments, we found that setting $r = 2$ and $k_1$ very close to, or smaller than, the size of the original parallel data consistently gives good results across language pairs. Fine-tuning $r$ and $k_1$ would result in a better translation quality but at a greater cost.

| #lines back-translated | #BLEU |
|---|---|
| 10M | 21.4 |
| 20M | 21.4 |
| 40M | 21.5 |

Table 4: Results for different sizes of back-translated data for the Zh→En translation direction on Newsdev2017.

both translation directions until the fourth iteration that back-translated 1.6M sentences (approximately 8 times the size of the original parallel data). In our preliminary experiments, we found that incremental training significantly improves the translation quality over an NMT system that was trained directly, on the same amount of back-translated sentences. For instance, we observed a 0.6 BLEU points improvements for Tr→En over a system trained on 1.6M sentences back-translated by a system trained on the original parallel data (as in (Sennrich et al., 2016a)).

### 4.2 Setting for Zh-En

For the Zh-En language pair, since much larger parallel data were provided to train the system, we did not perform the incremental back-translation described in Section 4.1. For En→Zh, we back-translated the entire XMU Chinese monolingual corpus containing 5.4M sentences as the source to produce synthetic English data. For Zh→En, we empirically compared the impact of back-translating different sizes of English monolingual data, using the first 10M, 20M, and 40M lines of the concatenation of News Crawl-2016 and News Crawl-2017 English corpora to produce synthetic Chinese data. As shown in Table 4, there is not a significant difference in exploiting back-translated data as large as 40M lines compared to only 10M lines. Therefore, we selected the first 10M lines of the News Crawl-2016 English corpus to produce synthetic Chinese data.

## 5 Combination of NMT and SMT

Although we can expect SMT to perform very poorly for all the language pairs we considered,[10] our primary submissions for WMT18 are the results of a simple combination of NMT and SMT. Indeed, as demonstrated by Marie and Fujita (2018), and despite the simplicity of the method used, combining NMT and SMT makes MT more robust and can significantly improve translation quality, even when SMT greatly underperforms NMT. Following Marie and Fujita (2018), our combination of NMT and SMT works as follows.

### 5.1 Generation of $n$-best Lists

We first produced the 100-best translation hypotheses with our NMT and SMT systems, independently.[11] Unlike `Moses`, `Marian` must use a beam of size $k$ to produce a $k$-best list during decoding. However, using a larger beam size during decoding for NMT may worsen translation quality (Koehn and Knowles, 2017).[12] Consequently, we also produced with `Marian` the 10-best lists, for Zh-En, and 12-best lists for the other language pairs, and merged them with `Marian`'s 100-best lists to obtain lists containing up to 110 or 112 hypotheses.[13] In this way, we make sure that we still have hypotheses of good quality in the lists despite using a larger beam size.[14] Then, we merged the lists produced by `Marian` and `Moses`. We rescored all the hypotheses in the resulting lists with a reranking framework using features to better model the fluency and the adequacy of each hy-

[10] Especially due to the rich morphology of the languages involved and the long distance reorderings to perform in order to produce a translation of good quality.

[11] We used the option `distinct` in `Moses` to avoid duplicated hypotheses, i.e., with the same content but obtained from different word alignments, and consequently to increase diversity in the generated $n$-best lists.

[12] For Zh-En, the decoding of the test data with $k$=100 resulted in a drop of 0.4 BLEU points compared to a decoding with $k$=10. However, for the other language pairs we did not observe such a quality drop but instead a consistent and slight improvement of BLEU scores.

[13] Note that we did not remove duplicated hypotheses that may appear, for instance, in both 10-best and 100-best lists.

[14] Note that we could have also generated many individual smaller $n$-best lists, for instance using all our NMT models independently, and merge them to increase the diversity of the hypotheses list to rerank and therefore obtained better results. However, we decided to leave the exploration of this possibility for feature work.

| Feature | Description |
|---|---|
| L2R (6) | Scores given by each of the 6 left-to-right Marian models |
| R2L (2) | Scores given by each of the 2 (or 4 for Tr-En) right-to-left Marian models |
| LEX (4) | Sentence-level translation probabilities, for both translation directions |
| LM (2) | Scores given by the two language models used by the Moses baseline systems |
| WPP (2) | Averaged word posterior probability |
| LEN (2) | Difference between the length of the source sentence and the length of the translation hypothesis, and its absolute value |
| SYS (1) | System flag, 1 if the hypothesis comes from Moses $n$-best list or 0 otherwise |
| MBR (2) | For Tr-En only: MBR decoding using sBLEU and chrF++ |
| PBFD (1) | For Tr-En only: The phrase-based forced decoding score |
| L2R-bwd (6) | Scores given by each of the 6 left-to-right Marian models for the backward translation direction |
| R2L-bwd (2) | Scores given by each of the 2 (or 4 for Tr-En) right-to-left Marian models for the backward translation direction |

Table 5: Set of features used by our reranking systems. The column "Feature" refers to the same feature name used in Marie and Fujita (2018). Note that the two last feature sets, "L2R-bwd" and "R2L-bwd," were not experimented in Marie and Fujita (2018). The numbers between parentheses indicate the number of scores in each feature set.

| # | System | Et→En | En→Et | Fi→En | En→Fi | Tr→En | En→Tr | Zh→En | En→Zh |
|---|---|---|---|---|---|---|---|---|---|
| 1. | Moses | 18.2 | 15.1 | 15.8 | 10.7 | 12.1 | 8.4 | 16.9 | 28.0 |
| 2. | Moses NMT-reranked | 20.2 | 17.6 | 17.5 | 12.2 | 14.2 | 10.1 | 19.0 | 29.9 |
| 3. | Marian single (w/o backtr) | 22.9 | 18.5 | 17.6 | 13.2 | 20.2 | 12.2 | 23.7 | 33.0 |
| 4. | Marian single (w/ backtr) | 28.6 | 24.0 | 23.1 | 16.8 | 25.2 | 18.0 | 24.7 | 37.2 |
| 5. | Marian ensemble (w/ backtr) | 29.1 | 24.3 | 23.6 | 17.3 | 25.8 | 18.3 | 25.9 | 37.9 |
| 6 | Moses + Marian | 30.7 | 25.2 | 24.9 | 18.2 | 26.9 | 19.2 | 26.7 | 39.7 |

Table 6: Detokenized BLEU-cased scores for our MT systems on the *Newstest2018* test set. "NMT-reranked" denotes the reranking of the Moses's 100-best hypotheses using all our NMT models (left-to-right and right-to-left, for both translation directions, trained with back-translated data) as features. "backtr" denotes the use or not of back-translated monolingual data. "Moses + Marian" denotes our combination of best NMT (#5) and SMT (#1) systems described in Section 5.

pothesis. This method can find a better hypothesis in these merged $n$-best lists than the one-best hypothesis originated by either Moses or Marian.

### 5.2 Reranking Framework and Features

We chose KB-MIRA as a rescoring framework and used a subset of the features proposed in Marie and Fujita (2018). As listed in Table 5, it includes the scores given by the 6 left-to-right NMT models used to perform ensemble decoding (see Section 3.1). We also used as features the scores given by right-to-left NMT models that we trained for each translation direction with the same parameters as left-to-right NMT models. The two right-to-left NMT models, each achieving the best BLEU and the best perplexity scores on the development data, were selected, giving us two other features for each translation direction. Since the Tr-En training parallel data are much smaller, we were able to perform one more right-to-left training run for Tr→En and En→Tr.[15] We also experimented with the use of the scores computed from the NMT models trained for the backward translation direction. In total, we have then 16 features, or 20 for Tr-En, computed from NMT models. All the following features we used are described in details by Marie and Fujita (2018). We computed sentence-level translation probabilities using the lexical translation probabilities learned by mgiza during the training of our SMT systems. The two language models trained for SMT for each translation direction were also used to score the $n$-best translation hypotheses. To account for hypotheses length, we added the difference, and its absolute value, between the number of tokens in the translation hypothesis and the source sentence. As a consensus-based feature, we used the word posterior probabilities.

For only the Tr-En language pair, we were also able to compute a phrase-based forced decoding

[15] In practice, adding one more right-to-left model for reranking did not significantly improve the BLEU score on the development data.

score (Zhang et al., 2017) thanks to the small size of the phrase table learned for this language pair. Also only for this language pair, we computed the scores for each hypothesis given by the so-called minimum Bayes risk (MBR) decoding for $n$-best list using two metrics: sBLEU and chrF++ (Popović, 2017).

The reranking framework was trained on $n$-best lists produced by the decoding of the same development data that we used to validate NMT system's training and to tune SMT's model weights.

## 6 Results

The results of our systems computed for the Newstest2018 test set are presented by Table 6.

As expected, SMT systems greatly underperformed our best NMT systems with differences in BLEU points ranging from 6.6 (En→Fi) to 13.7 (Tr→En). Reranking `Moses` 100-best hypotheses using NMT models (NMT-reranked) significantly improved the translation quality for all the translation directions. For Fi→En, `Moses` NMT-reranked performed only 0.1 BLEU points worse than `Marian` single (w/o backtr). This result demonstrates the ability of SMT in producing better translation hypotheses than its one-best hypothesis. Indeed, a better translation can be easily retrieved with the help of NMT models within the 100-best lists. Using back-translated data during training was very effective for Et-En, Fi-En, and Tr-En, with improvements ranging from 3.6 to 5.8 BLEU points. Improvements were less significant for Zh-En, especially for Zh→En with only 1.0 BLEU points of improvements. This may be explained by the much larger parallel data already used to train systems for Zh-En. Another interesting finding is the relative inefficiency of using an ensemble of 3 models for NMT decoding with the transformer architecture over using a single model, as opposed to what was reported by most participants at WMT17 (Bojar et al., 2017) using RNN. For instance, for En→Et and En→Tr ensemble decoding improved the translation quality by only 0.3 BLEU points.

Our combination of SMT and NMT significantly outperformed all our NMT systems for all translation directions. For instance, this combination brought 1.6 and 1.8 BLEU points of improvements for Et→En and En→Zh, respectively, over our best NMT systems.

## 7 Conclusion

We participated in eight translation directions and for all of them we did experiments to compare SMT and NMT performances. While SMT significantly underperforms NMT, we showed that a simple combination of both approaches delivers the best results.

## Acknowledgments

We would like to thank all the reviewers for their valuable comments and suggestions. This work was conducted under the program "Promotion of Global Communications Plan: Research, Development, and Social Demonstration of Multilingual Speech Translation Technology" of the Ministry of Internal Affairs and Communications (MIC), Japan.

## References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Colin Cherry and George Foster. 2012. 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, Montréal, Canada. Association for Computational Linguistics.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria. Association for Computational Linguistics.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine*

*Translation and Generation*, pages 18–24. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

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. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Benjamin Marie and Atsushi Fujita. 2018. A smorgasbord of features to combine phrase-based and neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 111–124. Association for Machine Translation in the Americas.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2017. Improving neural machine translation through phrase-based forced decoding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 152–162, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 555–562. Association for the Advancement of Artificial Intelligence.

# PROMT Systems for WMT 2018 Shared Translation Task

**Alexander Molchanov**
PROMT LLC
17E Uralskaya str. building 3, 199155,
St. Petersburg, Russia
Alexander.Molchanov@promt.ru

## Abstract

This paper describes the PROMT submissions for the WMT 2018 Shared News Translation Task. This year we participated only in the English-Russian language pair. We built two primary neural networks-based systems: 1) a pure Marian-based neural system and 2) a hybrid system which incorporates OpenNMT-based neural post-editing component into our RBMT engine. We also submitted pure rule-based translation (RBMT) for contrast. We show competitive results with both primary submissions which significantly outperform the RBMT baseline.

## 1 Introduction

This paper provides an overview of the PROMT submissions for the WMT 2018 Shared News Translation Task. This year we participate with neural MT systems for the first time. We participate only in the English-Russian language pair, but with three different systems.

The paper is organized as follows: Section 2 is a brief overview of the submitted systems. Section 3 describes the data preparation, preprocessing and statistics in detail. Section 4 provides a description of the systems. In Section 5 we present and discuss the results. Section 6 concludes the paper.

## 2 Systems overview

We submitted three systems for the WMT 2018 Shared News Translation Task:

- A (almost) pure NMT system based on the Marian (Junczys-Dowmunt et al., 2018) toolkit. The system features a rule-based names processing module and backoff to RBMT baseline in a few cases.
- A hybrid NMT system based on the PROMT RBMT engine with OpenNMT-based (Klein et al., 2017) neural post-editing module.
- pure RBMT system.

## 3 Data

We use the data provided by the WMT organizers, some private in-house news parallel data (approximately 600k parallel sentences crawled from various news web-sources and dated between 2015 and 2017) and the TED Talks corpus from the OPUS website (Tiedemann, 2012). The NewsCommentary, TED and in-house corpora are used as is.

We do not use any data for fine-tuning. We use the WMT newstest2017 set as our validation set. We also report results for newstest2018.

### 3.1 Data filtering

The CommonCrawl and (especially) ParaCrawl corpora were heavily filtered and normalized using the PROMT tools and algorithms (including language recognition, removal of meaningless sentences, in-house tools for parallel sentences classification, spellchecker etc.). We discarded roughly 50% of the CommonCrawl and 60% of the ParaCrawl data.

The MultiUN corpus was only checked for sentence length ratio using a simple rule-based algorithm. Less than 1% of the original data was discarded.

After that, we applied the bilingual data selection algorithm (Axelrod et al., 2011) to the filtered versions of ParaCrawl and MultiUN. We use the English and Russian news 2016-2017

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 456–460
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64047

corpora from statmt.org as the in-domain corpora. After this procedure we selected 1.5M sentences from the ParaCrawl corpus and 6M sentences from the MultiUN corpus.

The final statistics for the training data are shown in Table 1.

## 3.2 Data preprocessing

### Pure NMT system

We adopt a standard preprocessing scheme using the scripts provided by the Marian toolkit. The data is tokenized using the Moses toolkit (Koehn et al., 2007) tokenizer; after that we apply truecasing and, finally, byte pair encoding (BPE) (Sennrich et al., 2016) with 85K operations for source and target. We do not use a shared vocabulary due to the Cyrillic nature of Russian alphabet.

### Hybrid NMT system

We adopt a slightly different pipeline for the OpenNMT-based system. The data is tokenized with the OpenNMT tokenizer. The tokenizer provides a nice and handy option of applying the case feature, thus there is no need for truecasing. Then, we apply BPE with the same size 85K operations for source and target using the OpenNMT BPE script. The OpenNMT BPE learning algorithm is an extended version of the original BPE script adopted in Marian and has the following additional features: 1) the BPE merge operations are learnt to distinguish subword units at the beginning, in the middle and at the end of the word and 2) the BPE merge operations are learnt in case-insensitive mode (as we use the case feature to handle that). The OpenNMT system architecture does not support shared embeddings so despite the fact that both source (RBMT translations) and target (human translations) data is encoded in Cyrillic we train separate BPE models.

## 3.3 Synthetic data

We use three types of additional synthetic data described in detail below. The final size of the training data for the pure NMT system is roughly 4 times the total size of the filtered data in Table 1, while the final size of the training data for the hybrid system is approximately 6 times the size of the filtered data.

| **Corpus** | **#sent** | **#tokens EN** | **#tokens RU** |
|---|---|---|---|
| MultiUN | 6.0 | 140.8 | 129 |
| ParaCrawl | 1.5 | 28.4 | 24.3 |
| Yandex corpus | 0.6 | 16.8 | 15.4 |
| Private data | 0.6 | 15.6 | 15 |
| CommonCrawl | 0.4 | 10.3 | 9.5 |
| NewsCommentary | 0.3 | 6.2 | 5.9 |
| TED Talks | 0.1 | 2.4 | 2.1 |
| **Total** | **9.5** | **220.5** | **201.2** |

Table 1: Statistics for the filtered parallel English-Russian data in millions of sentences (#sent) and tokens.

### Back-translated data

Using the filtered data presented in Table 1 we train two initial auxiliary target-to-source NMT systems using the filtered data:

- A Russian-English NMT system using Marian (s2s with default parameters);
- A Russian-to-RBMT NMT system using OpenNMT (dbrnn, 2 layers, RNN size 1024 units).

The trained systems are then used to back-translate the 2017 news corpus from statmt.org (in case with the Marian system, we translate from Russian into English; the OpenNMT systems translates from Russian into the “Rule-based Russian”, mimicking the rule-based machine translation accent and structure). The size of the synthetic corpus is approximately equivalent to the size of human training data.

### Replicated data with unknown words

Similar to (Pinnis et al., 2017), we again roughly double our parallel data by creating a synthetic parallel corpus using the following steps: first, we perform word-alignment of our initial parallel training corpus using the MGIZA tool (Gao and Vogel, 2008). Then, we randomly replace from one to three unambiguously (one-to-one) aligned subword units in both source and target parallel sentences with the special <UNK> placeholder. The same pipeline is applied to both pure NMT system (for which we augment the English-Russian corpus) and the hybrid NMT system (for which we augment the RBMT-human Russian corpus) and to both the initial and back-translated data.

### Monolingual data

To benefit from the fact that we have data in Cyrillic in both source (RBMT) and target (human Russian) when dealing with the hybrid system, we add the 2017 Russian news corpus from statmt.org to the source side of the training data of the hybrid NMT system and replicate it on the target side. Currey et al. (2017) claim that this technique can yield improvements for translation of named entities. The BPE models learnt on the initial training data are applied.

## 4 Systems architecture

This section describes the trained systems in detail.

### 4.1 RBMT system

The PROMT RBMT System is a mature machine translation system with huge linguistic structured databases containing morphological, lexical and syntactic features for most European and Russian languages. We did not do any specific tuning for our submission.

### 4.2 Pure NMT system

For the pure NMT system we train a transformer (Vaswani et al., 2017) model. We use the recipe available at the Marian website[1]. The system configuration, hyperparameters and training steps follow those in the recipe. There are two minor differences: 1) we check the validation translation less frequently and set a higher early-stopping threshold to allow the model iterate over the training data for several epochs; 2) we do not use shared vocabulary because of the different alphabets in English and Russian. First, we trained the baseline system on the initial parallel data and back-translated data. After that, we trained 4 other models with different seeds using the whole data augmented with unknown words (see section 3.3).

### Model configuration

We use an ensemble of all 5 transformer models as our baseline translation system; in addition, we use RBMT as our back-off system (this will be described in detail in the next section). We use the beam of size 12 and the “normalize” parameter is set to 1.

### Back-off to RBMT

At first we had in mind training a classifier to choose when to fall back to the RBMT model. However, linguistic analysis of the neural translation of the validation set showed us that the NMT output is of good quality. We only encountered two minor problems: 1) the model sometimes outputs English text (less than 1% of the validation set sentences) and 2) from time to time the decoder outputs multiple recurring words or n-grams (this is a well-known problem of NMT systems). We deal with both problems using simple rules. First, the model output is checked using language recognition tool. If the language is other than the Russian, we fall back to the RBMT translation. Additionally, we check the neural translation for recurring words or n-grams: if a word recurs more than twice or an n-gram recurs more than once, we also fall back to the RBMT system.

### Handling proper names

We noticed that our transformer models have a problem translating proper names, especially rare ones or the ones not seen in the training data. Linguistic analysis led us to the conclusion that problems occur most often with the proper names which either 1) appear less than 5 times in the training corpus or 2) are split by the BPE model. To deal with this issue, we developed the following pipeline. We use the Stanford NER tool (Finkel et al., 2005) to identify proper names in the source text (person names, organizations and locations). We check the name frequency in the training data and whether it is split by the BPE model. If the frequency of any part of the name is low or it is split, we replace the whole name with the <UNK> placeholder in the source sentence. Then we translate the sentence by an ensemble of 4 models trained to reproduce unknown words allowing the decoder to reproduce unknown words in the output. Finally, we substitute the <UNK> placeholders in the output with the translations of the names produced by the RBMT system. If for some reason we can’t match the names to their RBMT translations or the number of the <UNK> placeholders in the NMT system output is not equal to the number of the placeholders in the source sentence, we fall back to the baseline NMT system described in Subsection 4.2 above.

[1] https://github.com/marian-nmt/marian-examples/tree/master/wmt2017-transformer

| Source sentence | NMT | NMT+names | Reference |
|---|---|---|---|
| The Russians represented in qualifying were Anton Chupkov, Evgeny Koptelov, Alexander Sukhorukov, and Grigory Tarasevich. | В квалификации россияне представляли Антона Чупакова, Евгения Коптева, Александра Сухокова и Григория Тараскевича. | В квалификации были представлены россияне Антон Чупков, Евгений Коптелов, Александр Сухоруков и Григорий Тарасевич. | Россиян в квалификации представили Антон Чупков, Евгений Коптелов, Александр Сухоруков и Григорий Тарасевич. |
| They all lived in the small town of Greenfield, Massachusetts. | Все они жили в небольшом городе Гринсфилл, штат Техас. | Все они жили в небольшом городе Гринфилд, Массачусетс. | Все они жили в небольшом городке Гринфилд в штате Массачусетс. |

Table 2. Examples of translation with names processing. The NMT+names stands for the system with proper names processing as described in Section 4.2.

### 4.3 Hybrid NMT system

We mentioned earlier that OpenNMT does not support the transformer model architecture. Due to this fact we train a model with a deep bidirectional encoder and a decoder with attention (Luong et al., 2015). Both encoder and decoder consist of two layers each with 1024 hidden units. The word embeddings size is 500 and the case feature embeddings size is 4. As with the pure NMT system, we first trained a baseline model on the initial parallel data and back-translated (Russian-to-RBMT) data. After that, we retrained the baseline model on the whole data augmented with unknown words and monolingual data (see Section 3.3 for details). We train the baseline model for 8 epochs and then retrain the model on all data for two more epochs. We use the beam of size 8 for translation.

Linguistic analysis of the translation of the validation set didn't show any problems regarding the NMT post-editing component. Thus, we made a decision not to make any special processing of names or fall back to RBMT and submit the hybrid post-edited translation as is.

## 5 Results and discussion

In this section we present the BLEU (Papineni et al., 2002) scores for our systems on two test sets and the linguistic analysis of the results.

The scores are presented in Table 3. Calculation is done using the `multi-bleu-detok.perl` script from the Moses toolkit.

We also studied the impact of the proper names processing applied to the NMT translation. Our pipeline affected 815 (27%) out of 3000 sentences in the test set. As we can see, unfortunately the BLEU is a bit lower than for the default translation. We see two reasons for that: first, we lose precision because frequently a name, even translated correctly, appears in the wrong case in the output. Russian is a highly inflective language and this is a problem. Marian does not support factored translation yet, so we couldn't teach the system to output the case feature for our placeholders. Secondly, the system was trained to reproduce placeholders for subword units and not the whole words, as we generated the synthetic data from the already BPE-segmented parallel bitexts. We chose, however, the translation with names processing to be our final submission as we decided that a system which is a little less fluent but more accurate at translating names would be better. Examples of translations with and without the names processing can be found in Table 2.

## 6 Conclusions and Future work

In this paper we have described our English-Russian submissions for the WMT 2018 Shared News Translation Task. Overall we have made three submissions: 1) a pure NMT system developed with the Marian toolkit, 2) a hybrid system with a NMT post-editing component

| System | newstest2017 | newstest2018 |
|---|---|---|
| RBMT | 22.9 | 18.1 |
| NMT | **31.0** | **27.4** |
| NMT+names | **30.9** | **27.3** |
| Hybrid | 29.5 | 25.3 |

Table 3: Results for the submitted systems. The NMT+names stands for the system with proper names processing as described in Section 4.2.

trained with the OpenNMT toolkit, and 3) pure RBMT system.

The pure NMT system with the state-of-the-art transformer architecture proved to be the best among our submissions in terms of BLEU.

We also present a names processing and translation pipeline which can be improved by teaching the system to output the translations in the correct case.

## References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–362, Edinburgh, Scotland, UK.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, Ann Arbor, MI, USA.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing, SETQA-NLP 08*, pages 49–57, Stroudsburg, PA, USA.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *Computing Research Repository*, arXiv:1701.02810. Version 2.

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. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL 07*, pages 177–180, Stroudsburg, PA, USA.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (2015)* pages 1412–1421, Lisbon, Portugal.

Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 02)*, pages 311–318, Philadelphia, PA, USA.

Marcis Pinnis, Rihards Krišlauks, Daiga Deksne, and Toms Miks. 2017. Neural Machine Translation for Morphologically Rich Languages with Improved Sub-word Units and Synthetic Data. In *Proceedings of the 20th International Conference of Text, Speech and Dialogue (TSD2017)*, pages 237–245, Prague, Czechia.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. 2016. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (2016)*, pages 35–40, San Diego, CA, USA.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *Proceedings of The North American Chapter of the 342 Association for Computational Linguistics Conference (NAACL-07)*, pages 508–515, Rochester, NY, USA.

Jorg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA.

# NTT's Neural Machine Translation Systems for WMT 2018

**Makoto Morishita, Jun Suzuki**[*] and **Masaaki Nagata**
NTT Communication Science Laboratories, NTT Corporation, Japan
{morishita.makoto, nagata.masaaki}@lab.ntt.co.jp
jun.suzuki@ecei.tohoku.ac.jp

## Abstract

This paper describes NTT's neural machine translation systems submitted to the WMT 2018 English-German and German-English news translation tasks. Our submission has three main components: the Transformer model, corpus cleaning, and right-to-left $n$-best re-ranking techniques. Through our experiments, we identified two keys for improving accuracy: filtering noisy training sentences and right-to-left re-ranking. We also found that the Transformer model requires more training data than the RNN-based model, and the RNN-based model sometimes achieves better accuracy than the Transformer model when the corpus is small.

## 1 Introduction

This paper describes NTT's submission to the WMT 2018 news translation task (Bojar et al., 2018). This year, we participated in English-to-German (En-De) and German-to-English (De-En) translation tasks. The starting point of our system is the Transformer model (Vaswani et al., 2017), which recently established better performance than conventional RNN-based models (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015). We incorporated a parallel corpus cleaning technique (Section 3.1) and a right-to-left $n$-best re-ranking technique (Section 3.4) and also used a synthetic corpus to exploit monolingual data. To maintain the quality of the synthetic corpus, we checked its back-translation BLEU scores and filtered out the noisy data with low scores (Section 3.2).

Through experiments, we evaluated how each feature affects accuracy (Section 4). Compared with the RNN-based system, we also identified when the Transformer model works effectively (Section 4.3.3).

## 2 Neural Machine Translation

Neural Machine Translation (NMT) has been making rapid progress in recent years. Sutskever et al. (2014) proposed the first NMT model that uses a simple RNN-based encoder-decoder network. Luong et al. (2015); Bahdanau et al. (2015) augmented this architecture with an attention mechanism, allowing the decoder to refer back to the encoder-side information at each time step. These conventional NMT models use RNNs as encoder and decoder to model sentence-level information. However, the RNN-based model uses previous states for predicting subsequent target words, which can cause a bottleneck in efficiency. Recently, Vaswani et al. (2017) proposed a model called Transformer, which completely relies on attention and feed-forward layers instead of RNN architecture. This model enables evaluation of a sentence in parallel by removing recurrence in the encoder/decoder, and we can train the model significantly faster than RNN-based models. It also established a new state-of-the-art performance in WMT 2014 translation tasks while shortening the training time by its GPU efficient architecture. In preliminary experiments, we also confirmed that the Transformer model tends to achieve better accuracy than RNN-based models, and thus we changed our base model for 2018 to the Transformer. For further details and formulation on the Transformer model, see Vaswani et al. (2017).

## 3 System Features

This year's submission includes the following features:

- Noisy data filtering for Common Crawl and ParaCrawl corpora (Section 3.1).
- Synthetic parallel data from the monolingual corpus (News Crawl 2017) with

[*] His current affiliation is Tohoku University.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 461–466
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64048

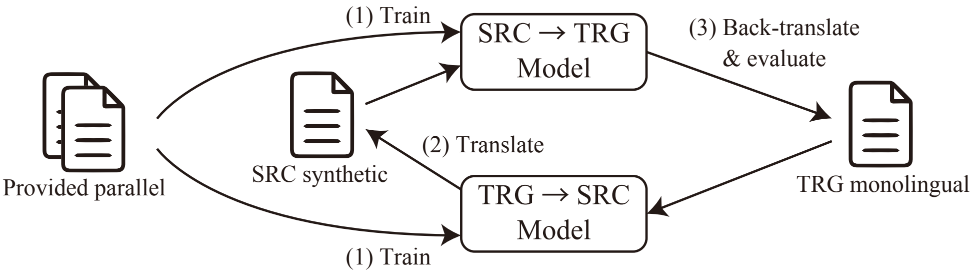


Figure 1: Overview of back-translation BLEU-based synthetic corpus filtering

back-translation BLEU-based filtering (Section 3.2).

- $n$-best re-ranking by a right-to-left translation model (Section 3.4).

From here, we discuss these features and experimentally verify each one.

### 3.1 Noisy Data Filtering

This year, ParaCrawl and Common Crawl corpora, which were created by crawling parallel websites, were provided for training. Since these web-based corpora are large but noisy, it seems essential to filter out noisy sentence pairs. Since the ParaCrawl corpus has already been cleaned by Zipporah (Xu and Koehn, 2017), we chose another method for further cleaning[1].

To clean the corpus, we selected the `qe-clean`[2] toolkit (Denkowski et al., 2012), which uses a language model to evaluate a sentences naturalness and a word alignment model to check whether the sentence pair has the same meaning. Both models are trained with clean data for scoring possibly noisy parallel sentence pairs and removes sentences with scores below a threshold. For more details, see Denkowski et al. (2012).

We used Europarl, News Commentary, and Rapid corpora as clean parallel data for training the word alignment model. We also used News Crawl 2017 as an additional monolingual corpus for language modeling. Since our target is news translation, using a news-related monolingual corpus is beneficial to train language models. We used `KenLM` (Heafield, 2011) and `fast_align` (Dyer et al., 2013, 2010) for language modeling and word alignment. To find the appropriate weights for each feature, we used newstest 2017 as a development set and fixed the threshold as one standard deviation.

### 3.2 Synthetic Corpus

One drawback of NMT is that it can only be trained with parallel data. Using synthetic corpora, which are pseudo-parallel corpora created by translating monolingual data with an existing NMT model, is one of the ways to make use of monolingual data (Sennrich et al., 2016a). We created a synthetic corpus by translating monolingual sentences with a target-to-source translation model and used it as additional parallel data.

In our case, we trained a baseline NMT model with a provided parallel corpora[3] and translated News Crawl 2017 to make a synthetic corpus.

### 3.3 Back-translation BLEU-based Filtering for Synthetic Corpus

A synthetic corpus might contain noise due to translation errors. Since these noisy sentences might deleteriously affect the training, we filtered them out.

In this work, we did back-translation BLEU-based synthetic corpus filtering (Imankulova et al., 2017). We hypothesize that synthetic sentence pairs can be correctly back-translated to the target language unless they contains translation errors. Based on this hypothesis, we found better synthetic sentence pairs by evaluating how the back-translated sentences resembled the original source sentences.

Figure 1 shows an overview of our synthetic corpus filtering process. First, we trained the NMT model with the provided parallel corpora and then translated the monolingual sentences in the target language to the source language by a target-to-

[1] Although the provided ParaCrawl corpus was already filtered by Zipporah (Xu and Koehn, 2017), a cursory glance suggested that it still contains many noisy sentence pairs.

[2] https://github.com/cmu-mtlab/qe-clean

[3] Europarl + News Commentary + Rapid + a filtered version of Common Crawl and ParaCrawl corpora

source translation model. After getting the translation, we back-translated it with the source-to-target model. Then we evaluated how well it restored the original sentences by sentence-level BLEU scores (Lin and Och, 2004), selected the high-scoring sentence pairs, and created a synthetic corpus whose size equals the naturally occurring parallel corpus.

### 3.4 Right-to-Left Re-ranking

Liu et al. (2016) pointed out that RNN-based sequence generation models lack reliability when decoding the end of the sentence. This is due to its autoregressive architecture that uses previous predictions as context information. If the model makes a mistake, this error acts as a context for additional predictions, often causing further errors.

To alleviate this problem, Liu et al. (2016) proposed a method that re-ranks an $n$-best hypothesis generated by the Left-to-Right (L2R) model, which generates a sentence from its beginning (left) to its end (right), by the Right-to-Left (R2L) model that generates a sentence in the opposite order. Their work mainly focuses on the problem of RNN-based models and the effect is unclear when applied to the Transformer model, which completely relies on attention and feed-forward layers. We assume this method also works with the Transformer model because it still has autoregressive architecture in its decoding phase.

We re-ranked the $n$-best hypothesis of the L2R model by the R2L model with the following formula:

$$P(\tilde{y}) = \arg\max_{y \in \boldsymbol{Y}} P(y|x;\theta_{L2R})P(y^r|x;\theta_{R2L}), \tag{1}$$

where $\boldsymbol{Y}$ is a set of $n$-best translations of source sentence $x$ obtained by the L2R model, $y^r$ is a reversed sentence of $y$, and $\theta_{L2R}$ and $\theta_{R2L}$ are the model parameters for the L2R and R2L models, respectively. In our experiments, we set $n = 10$.

## 4 Experiments

### 4.1 Data

As the first step of our data preparation, we applied the moses-tokenizer[4] and the truecaser[5] to all the datasets used in our experiments. Then we split the words into subwords by joint Byte-Pair-Encoding (BPE) (Sennrich et al., 2016b) with 32,000 merge operations. Finally, we discarded from the training data the sentence pairs that exceed 80 subwords either in the source or target sentences. As a development set, we used newstest 2017 (3004 sentences).

### 4.2 Translation model

**Transformer** We used the `tensor2tensor`[6] implementation to train the Transformer model. Our hyper-parameters are based on the previously introduced Transformer big setting (Vaswani et al., 2017), and we also referred Popel and Bojar (2018) for tuning hyper-parameters. We used six layers for both the encoder and the decoder. All the sub-layers and the embeddings layers output 1024 dimension vectors, and the inner-layer of the position-wise feed-forward layers has 4096 dimensions. For multi-head attention, we used 16 parallel attention layers. We use the same weights for the encoder/decoder embedding layers and the decoder output layer by three-way-weight-tying (Press and Wolf, 2017). As an optimizer, we used Adam (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.997$ and set dropout (Srivastava et al., 2014) with a probability of 0.1. We used a learning rate decaying method proposed by (Vaswani et al., 2017) with 16,000 warm-up steps and trained the model for 300,000 steps. Each mini-batch contained roughly 20,000 tokens. We saved a model every hour and averaged the last 16 model parameters for decoding. The training took about three days for both En-De and De-En with eight GTX 1080Ti GPUs. During decoding, we used a beam search with a size of ten and a length normalization technique (Wu et al., 2016) with $\alpha = 1.0$ and $\beta = 0.0$.

**RNN-based** In several experimental settings, we also trained an RNN-based attentional NMT model based on a previous work (Luong et al., 2015) for comparison[7]. We used a two-layer LSTM-based model and respectively set the embedding and hidden layer unit sizes to 512 and 1024. As an optimizer, we used SGD and set an initial learning rate to 1.0. We decayed the learn-

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

[5] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/recaser/truecase.perl`

[6] `https://github.com/tensorflow/tensor2tensor`

[7] Implementation and settings are based on our submission to WAT shared-task (Morishita et al., 2017).

ing rate after 13 epochs by multiplying 0.7 per epoch and trained the model for 20 epochs. We clipped the gradient (Pascanu et al., 2013) if its norm exceeded 5.0. We set the dropout probability to 0.3. Each mini-batch contained about 128 sentences. The training took about 23 days for De-En and 31 days for En-De on a single GTX 1080Ti GPU. During decoding, we set the beam size to 20 and normalized the scores by dividing them by the sentence length.

### 4.3 Experimental Results and Discussions

Table 1 shows the provided and filtered corpus sizes for training. The Original Common Crawl and ParaCrawl corpora contain around 35.56M sentences. However, since most of the sentence pairs are noisy, we only retained the cleanest 4.01M sentences that were selected by the `qe-clean` toolkit. For the synthetic corpus, we chose the same size as the filtered parallel corpus based on the back-translation BLEU+1 scores.

Table 2 shows the evaluation results of our submission and baseline systems. Here, we report the case-sensitive BLEU scores (Papineni et al., 2002) evaluated by the provided automatic evaluation system[8]. In the following, unless specified, we mainly discuss the Transformer model results.

#### 4.3.1 Effect of Corpus Filtering

We split the provided corpora into two parts: (1) Europarl, News Commentary and Rapid corpora as clean, and (2) Common Crawl and ParaCrawl corpora as noisy.

First, we just trained the model with cleaner corpora (Setting `(1)`) and added possibly noisy corpora (Setting `(2)`). The noisy parallel corpus seriously damaged the model for En-De, although there was a small gain for De-En. After filtering out the noisy part of the corpora (Setting `(3)`, it showed a large gain of +11.3 points for En-De and +4.8 points for De-En compared to the unfiltered setting. This suggests that clean, small training data tend to outperform large but noisy data. This large gain might also come from the effect of domain adaptation. We used news-related monolingual sentences to train the language model for corpus filtering, and thus our filtered sentences are related to a news domain, which is the same as our test set.

Then we added a synthetic corpus with and without filtering (Settings `(4)` and `(5)`). Although adding an unfiltered corpus resulted in certain gain, we identified an additional gain of +3.5 points for En-De by filtering out low-quality synthetic sentence pairs based on back-translation BLEU+1 scores.

Synthetic corpus filtering worked well, especially for En-De; but we did not see a large difference for De-En. To determine why, we estimated the quality of the synthetic corpus by checking the back-translation BLEU+1 scores. Table 3 shows the average back-translation BLEU+1 scores of the filtered/unfiltered synthetic corpus. These scores reflect the translation accuracy of the synthetic sentences. Before filtering, the average En-De score was lower than the average De-En score. From this result, we suspect that De-Ens unfiltered synthetic corpus is clean enough, resulting in no improvement from further filtering. After choosing high-scoring sentence pairs, the average scores exceed 80 for both language pairs, ensuring the quality of the synthetic corpus.

From our experiments, we confirmed that noisy parallel sentence pairs significantly damaged the model. For the best results, noisy sentences must be filtered out before training the model.

#### 4.3.2 Effect of Right-to-Left Re-ranking

By re-ranking the $n$-best hypothesis by the R2L model, we saw a gain of 1.5 points for En-De and 0.5 points for De-En (Setting `(6)`). We submitted these results as our primary submission.

R2L $n$-best re-ranking works well with the RNN-based model, but we confirmed that it also works well with the Transformer model. We suppose both the Transformer and the RNN models lack the ability to decode the end of the sentence, but R2L model re-ranking can alleviate this problem.

#### 4.3.3 Comparison of Transformer and RNN

For settings `(1)`, `(3)`, and `(5)`, we also trained the RNN-based NMT for comparison. We compared the Transformer and the RNN and found the latter achieved comparable or sometimes better results than the Transformer when trained with a small parallel corpus (Settings `(1)` and `(3)`). When the corpus size increased after adding a synthetic corpus, Transformer surpassed the RNN (Setting `(5)`). Our results suggest that Transformer gets stronger when the parallel corpus is enough large, but it might be worse than the

[8] http://matrix.statmt.org/

| Corpus | Sentences |
|---|---|
| Europarl + News Commentary + Rapid | 3.10M |
| Common Crawl + ParaCrawl | 35.56M |
| Filtered version of Common Crawl + ParaCrawl | 4.01M |
| Synthetic corpus (News Crawl 2017) | 37.94M (En-De), 25.86M (De-En) |
| Filtered version of synthetic corpus (News Crawl 2017) | 7.11M |

Table 1: Number of sentences in datasets

| | | En-De | | | De-En | | |
|---|---|---|---|---|---|---|---|
| | Settings | Sentences | Transformer | RNN | Sentences | Transformer | RNN |
| (1) | Europarl + News Commentary + Rapid | 3.10M | 32.5 | 30.4 | 3.10M | 31.0 | 31.0 |
| (2) | (1) + Unfiltered Common Crawl + ParaCrawl | 38.66M | 26.6 | — | 38.66M | 32.7 | — |
| (3) | (1) + Filtered Common Crawl + ParaCrawl | 7.11M | 37.9 | 39.6 | 7.11M | 37.5 | 39.6 |
| (4) | (3) + Unfiltered synthetic corpus | 45.05M | 41.5 | — | 32.97M | 46.4 | — |
| (5) | (3) + Filtered synthetic corpus | 14.22M | 45.0 | 39.8 | 14.22M | 46.3 | 43.7 |
| (6) | (5) + R2L re-ranking (submission) | 14.22M | 46.5 | — | 14.22M | 46.8 | — |

Table 2: Cased BLEU scores of our submission and baseline systems

| | En-De | De-En |
|---|---|---|
| Unfiltered | 44.02 | 53.96 |
| Filtered | 80.12 | 80.81 |

Table 3: Average back-translation BLEU+1 scores of synthetic corpus

RNN-based models when the corpus size is small. One critical reason is that Transformer has many trainable parameters, complicating training with small training data. This result might change with smaller hyper-parameter settings (e.g., Transformer base setting), but we set aside this idea for future work.

## 5 Conclusion

In this paper, we described our submission to the WMT 2018 news translation task. Through experiments, we found that careful parallel corpus cleaning for the provided and synthetic corpora largely improved accuracy, and we confirmed that R2L re-ranking works well even with the Transformer model. Our comparison between the Transformer and RNN-based models suggests that the latter models might surpass the former when the training data are not enough large. This result sheds light on the importance of large, clean data for training the Transformer model.

## Acknowledgments

We thank two anonymous reviewers for their insightful comments and suggestions.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the 3rd Conference on Machine Translation (WMT)*.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The cmu-avenue french-english translation system. In *Proceedings of the 7th Workshop on Statistical Machine Translation (WMT)*, pages 261–266.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, pages 644–648.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7–12.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation (WMT)*, pages 187–197.

Aizhan Imankulova, Takayuki Sato, and Mamoru Komachi. 2017. Improving low-resource neural machine translation with filtered pseudo-parallel corpus. In *Proceedings of the 4th Workshop on Asian Translation (WAT)*, pages 70–78.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Chin-Yew Lin and Franz Josef Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 501–507.

Lemao Liu, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2016. Agreement on target-bidirectional lstms for sequence-to-sequence learning. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2630–2637.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2017. NTT neural machine translation systems at WAT 2017. In *Proceedings of the 4th Workshop on Asian Translation (WAT)*, pages 89–94.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, volume 28, pages 1310–1318.

Martin Popel and Ondřej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110:43–70.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th European Chapter of the Association for Computational Linguistics (EACL)*, pages 157–163.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1715–1725.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*, pages 6000–6010.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2945–2950.

# The Karlsruhe Institute of Technology Systems for the News Translation Task in WMT 2018

**Ngoc-Quan Pham** and **Jan Niehues** and **Alex Waibel**
Karlsruhe Institute of Technology
ngoc.pham@kit.edu jan.niehues@kit.edu alex.waibel@kit.edu

## Abstract

We present our experiments in the scope of the news translation task in WMT 2018, in directions: English→German. The core of our systems is the encoder-decoder based neural machine translation models using the transformer architecture. We enhanced the model with a deeper architecture. By using techniques to limit the memory consumption, we were able to train models that are 4 times larger on one GPU and improve the performance by 1.2 BLEU points. Furthermore, we performed sentence selection for the newly available ParaCrawl corpus. Thereby, we could improve the effectiveness of the corpus by 0.5 BLEU points.

## 1 Introduction

This manuscript provides the technical details regarding our submission in the WMT18 shared task on English→German news translation. Our submission has two major research contributions: Firstly, the development of a deep, efficient neural architectures and secondly, the cleaning and data selection of web crawled data.

We developed a efficient approach to train a deep transformer model on a single GPU. This allows use to train a 4 times deeper model than state-of-the-art models on one GPUs. In the experiments we are able to show that these models perform 1.2 BLEU points better than the baseline model using already 8 layers.

Secondly, we performed additional filtering on the ParaCrawl corpus. We are using the log-probabilities of a baseline NMT system to filter the low quality translations. While we are only able to improve the translation quality slightly by 0.3 BLUE points using all ParaCrawl data, the integration of the clean cropus improved the translation quality of 0.8 BLEU points.

## 2 Data

This section describes the preprocessing steps for the parallel and monolingual corpora for the language pairs involved in the systems as well as the data selection methods investigated.

### 2.1 English↔German

As parallel data for our German↔English systems, we used Europarl v7 (EPPS), News Commentary v12 (NC), Rapid corpus of EU press releases, Common Crawl corpus, the ParaCrawl corpus and simulated data. The preprocessing includes tokenization, removing very long sentences and the sentence pairs which are length-mismatched, normalizing special symbols and different writing rules and smart-casing the first word of each sentence. Those tools are provided in the Moses Toolkit [1].

We integrated the monolingual news data by generating synthetic data as motivated by Sennrich et al. (2016a). We used the translated data provided by University of Edinburgh.

Once the data is preprocessed, we applied byte-pair encoding (BPE) (Sennrich et al., 2016b) on the corpus. In this work, we deploy an operation size of 40K (shared between English and German languages) and applied vocabulary filtering in a way that every token occurs at least 50 times.

#### 2.1.1 ParaCrawl data selection

This year, in addition to the data provided in the last years, also the ParaCrawl corpus was provided. Since this data is collected by a web-crawler it differs in several ways from the other corpus. Firstly, it is significantly larger than all other available corpora. But the corpus is also more noisy. Therefore, we did not directly use this corpus, but filtered it prior to training.

[1] http://www.statmt.org/moses/

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 467–472
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64049

In this use case, an NMT system trained on the clean parallel data was evaluated. Therefore, we investigate the usage of this system to select clean translations from the training data.

In a first step, we performed the same preprocessing as for the other corpora. In addition, we removed short sentences. We noticed that these were often only keywords or numbers and therefore would not be helpful to train the system. In our experiments, we removed all sentences shorter than $n = 10$ words.

In the second step, we use the NMT system to calculate the translation probability of the ParaCrawl data. We used the length normalized log probability to select the sentences used for training. An inspection on a tiny subset of the data showed that the sentences with a low length-normalized probability seem to be bad translations. Examples are shown in the top of Table 1. Often the they are even not sentences in the source and target language. Furthermore, we noticed that the sentences with a very high probability seem not to be very useful. As shown in the last example in 1, in these cases, we often have a one-to-one word correspondence between the source and target sentence. But the input are often no real sentences and therefore, we might learn to generate no longer fluent output.

Due to computation time, we were not able to train model on different selected parts of the corpus. In contrast, we select reasonable thresholds based on the ordering on a small subset. We removed all sentences, where the length-normalized log-probability is smaller then $a = 0.8$ and all sentences where this score is higher than $b = 3.6$.

## 3 Deep Transformer

The research in Machine Translation has observed rapid advancement in terms of modeling in the past three years. While recurrent neural networks remain the core component in many strong systems (Wu et al., 2016), various works incrementally discovered that other architectures can also outperform RNNs in terms of translation quality or training efficiency, such as Convolutional Neural Networks (CNNs) (Gehring et al., 2017) or Self-Attention Networks, or Transformer (Vaswani et al., 2017). Due to the success of the self-attention networks, we will concentrate in this work on this type of architecture.

While other areas of deep learning use very deep neural networks, the networks used for NMT are still shallow compared to these areas. Motivated by the success of deep models in other areas, we analyzed the effectiveness of depth of the Transformer network. This is only possible trough the development of a very efficient implementation. This enables us to training very deep networks on a single device in a reasonable amount of time.

### 3.1 Sequence-to-Sequence models

Neural machine translation (NMT) consists of an encoder and a decoder (Sutskever et al., 2014; Cho et al., 2014) that directly approximate the conditional probability of a target sequence $Y = y_1, y_2, \cdots, y_T$ given a source sequence $X = x_1, x_2, \cdots, x_M$. The basic concept of the model is to encode the source sequence with a neural network to capture the neural representation of the source sentence, which is then referred multiple times during a decoding process, in which another neural network auto-regressively generates tokens in the target language.

The architectural choice is important in building neural machine translation systems. While Recurrent Neural Networks (RNN) have become the defacto model to represent sequences and were applied very successfully in NMT (Sutskever et al., 2014; Luong and Manning, 2015), self-attention networks (or Transformer) arose as a potentially better alternative (Vaswani et al., 2017).

### 3.2 Transformer overview

The transformer architecture was previously introduced with the following novel features:

- Long range dependency is modeled using the self-attention mechanism instead of recurrent connections used in recurrent networks, like the Long-Short Term Memories. The mechanism allows direct connection between two different two arbitrary positions in the sequences, which in turns alleviates the gradient flow problem existing in recurrent networks.

- Residual block design: similar to the infamous residual networks consisting of deep convolutional neural networks, Transformer networks are built on residual blocks in which the lower level states are directly carried to the top level by addiction. In the

| | |
|---|---|
| German: | offener Teilnahmewettbewerb : Grafikdesign fr Musikprojekt |
| English: | DAS Hotel |
| German: | anderen Gewinnen . |
| English: | Anyway , I will repeat that I sincerely hope you weren 't referring to me |
| German: | Christijan Albers 2 : 2 ( 3 : 2 im Elfmeterschieen ) |
| English: | Christijan Albers 2 : 2 ( 3 : 2 in penalty shootout ) |

Table 1: Filtered examples

Transformer networks, the input of every sub-block is added directly to the output (He et al., 2016), as a result the final layer receives a large sum of inputs from below, including the embeddings.

- Multi-head attention being proposed as a variation of the attention network (Bahdanau et al., 2014) improves attention power by performing attention in multiple dimensions of the input, which are projected using linear transformation.

- Additional neural network training utilities: layer normalization (Ba et al., 2016) prevents network state values from exploding; label smoothing regularizes the cross entropy loss function to improves the models' generalization;

### 3.3 Efficient memory usage

NMT models in general are very memory consuming due to the fact that they need to apply transformation on a sequence of states instead of single states in feed-forward neural networks. For other architectures, like feed-forward neural networks, convolution neural networks and recurrent neural networks, recently techniques have been proposed to significantly reduce the memory footprint during training (Chen et al., 2016; Gruslys et al., 2016). The main idea is to recalculate intermediate results instead of caching them. In this work, we adopted this idea to transformer models. We apply the method for a layer basis, by specifying the number of layers (Transformer Encoder or Decoder block) to be checkpointed during training. Such layer's forward pass needs to be recomputed during the backward pass, as a result the intermediate buffers created during training can be discarded, resulting in smaller memory requirement and bigger batch size.

### 3.4 Training

We followed the original work for the general hyper parameters including batch size and learning rate. We instead focus on several methods to increase training efficiency of the Transformer models.

**Emulated Multi-GPU setup**: It is notable that the *Noam* learning rate schedule proposed in (Vaswani et al., 2017) was designed for bigger batch sizes ($\approx$ 25000 words per mini-batch update which is not feasible for a single-GPU setup). In order to apply the same learning schedule without a multi-GPU system, we simply divide the large mini-batch into smaller ones, and accumulate (by summing) the gradients computed by each mini-batch forward and backward pass.

## 4 Results

### 4.1 Baseline System

Our baseline system uses the openNMT-py Toolkit[2] and uses an RNN based translation model with 4 layers in both decoders and encoders (bidirectional RNN on the encoder side). The model is equipped with dropout$= 0.2$ following the work of (Zaremba et al., 2014) for better regularization and label smoothing improving the cross-entropy loss. The training details and hyper-parameters are replicated from (Pham, 2017). In all of our experiments, we use the concatenation of test sets from 2013 to 2016 as our development set for model/checkpoint selection. While we use perplexity for model selection, the BLEU score on newstest2017 calculated by mteval-v13a.pl is used to report the models' performance.

### 4.2 Training hyper parameters

For RNN models, we use 4-layer-models with Long-Short Term Memory (Hochreiter and Schmidhuber, 1997). The bi-directional LSTM is used in the Encoder for all 4 layers. We use batch

[2] http://opennmt.net

size of 128 sentences (notably, the measurement of batch size in Transformer is denoted by the number of tokens, not sentences) and simply trained with Stochastic Gradient Descent with learning rate decay when the validation perplexity does not improve (Luong et al., 2015).

For Transformer models, we set the base layer size to 512, while the hidden layer in each Position Wise Feed Forward network has 2048 neurons, which matches the *Base* model in (Vaswani et al., 2017).

The learning method is Adam (Kingma and Ba, 2014) with the learning rate schedule similar to the original paper, with a minor difference that we increase the number of warm up steps to 8192 and double the base learning rate. If Dropout is applied, we use dropout at each Position Wise Feed Forward hidden layer and the attention weights.

### 4.3 Model comparison

In a first series of experiment we compared different architectures (RNNs and Transformers) and the influence of the deeps of the network. The transformer-based models are implemented using PyTorch (Paszke et al., 2017) and the source codes are open sourced. [3]. We provided our starting point as a reference to our participation to the last year's shared task. Thus, we use the corpus consisting of the Europarl, News Commentary, Rapid Corpus and the cleaned Common Crawl, which is then boosted with the back translation data provided by University of Edinburgh. The total data size is around 9 million sentence pairs.

| Model | BLEU (newstest2017) |
|---|---|
| Baseline (RNN) | 27.4 |
| Transformer-4 | 27.8 |
| Transformer-12 | 29.2 |
| Transformer-24 | 29.7 |

Table 2: RNNs vs Transformers (various depths) trained without paraCrawl.

As the results in Table 2 suggest, the baseline model despite having larger model size (1024) and being improved with dropout and label smoothing is not able to outperform a base Transformer (hidden size 512 for every layer) with only 4 layers. More importantly, the result scales over the Transformer's depth, such subject will be covered in the subsequent section. We managed to outperform the RNN baseline by 2.3 BLEU points just by increasing the depth to 24 layers.

Though we do not provide any comparison with respect to depth in Recurrent Neural Networks, previous work (Britz et al., 2017) explores different depths during training NMT models with similar architectures to our baseline discovering that it is not trivial to improve Recurrent NMT models just by increasing depth even with residual connections. It is notable that recent work (Chen et al., 2018) empirically proved that RNN models with hyper parameter tuning and layer normalization strategy can perform on par with the Transformer.

### 4.4 Data Size

As illustrated above, the Transformer models produced strong results which can outperform the best system of last year which is an ensemble of RNN models (Sennrich et al., 2017). We proceed to improve the system further by providing additional training data. Table 3 shows that a naive addiction of the paraCrawl data yields only a boost of 0.3 BLEU points, while our filtering method impressively improves the result by 0.8.

| Data | News2017 |
|---|---|
| Transformer-12 | 29.2 |
| +paraCrawl | 29.5 |
| + filtered paraCrawl | 30.0 |

Table 3: Experiments using different data sizes

### 4.5 When do we need regularization

Deeper models are more likely to overfit, which can be alleviated by using Dropout, specifically in the Position-wise feed forward network in each transformer block. We apply dropout at the the embeddings, residual connections (the output of the transformations before addiction) and at the attention matrices with the same probability of 0.1) The results in table 4 shows that Dropout started to be effective when the model becomes deeper than 12, even though the difference in the 16 configuration is rather subtle. At 12 layers and below, dropout seems to be unnecessary, possibly because our corpus size has reach 40 million sentences (included the filtered paraCrawl corpus).

Since we used the training regime which stops after $100K$ steps (each updates the parameter based on the batch size of about 25000 words), it is possible that Dropout models requires training

[3] https://github.com/isl-mt/NMTGMinor

for more than such threshold, due to the fact that a side effect of Dropout is to prolong the training progress.

### 4.6 Deeper networks

To answer the empirical question if very deep networks can improve the translation performance given abundant training data (as we have three times more data than the first experiment w.r.t depth), we managed to train networks as deep as 32 layers. The results are shown in Table 5. We observe significance improvement (0.7 BLEU points) in the first incremental steps from 4 to 12. The progress becomes stagnant from 16, and not until reaching 32 layers did we manage to obtain an additional 0.4 increase. The Transformer network clearly benefits from depth, which was not observed in Recurrent Network (Britz et al., 2017), however the effect is diminishing at 12 layers, while training models as deep as 32 is not simple. To the best of our knowledge, our model consists of totally 96+48+2 sub-blocks (encoders, decoders and input/output layers) which is the first attempt to explore a network with this depth in Neural Machine Translation.

Our training time ranges from 1 week with the 12-layer models to maximum of 2 weeks for the 32-layer models using single GTX 1080Ti graphics cards.

### 4.7 Final submission

The final submission of KIT is the ensemble of 5 models using different layer sizes and switching on and of dropout. Each of the models is already an average of different checkpoints. The results are summarized in Table 6. We found that the an ensemble of 5 models is only able to increase the score by 0.3, which shows that the 32-layer model dominates others.

## 5 Conclusion

In conclusion, we described our experiments in the news translation task in WMT 2018. The main focus of our submission was on data selection and techniques to efficient train deep transformer models . While we were only able to improve the translation performance slightly by using the whole ParaCrawl corpus, we could improve the translation performance by 0.8 BLEU points when using a filtered version of the corpus. We successfully filtered the data by using the translation probabilities of a baseline NMT system. Secondly, we were successfully in training a deep transformer model on a single GPU. By increasing the depth of the network by a factor of 4, we were able to gain additional 1.2 BLEU points. This was only possible by caching less data during training and recalculating them if needed.

## References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

D. Bahdanau, K. Cho, and Y. Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia. PMLR.

Audrunas Gruslys, Rémi Munos, Ivo Danihelka, Marc Lanctot, and Alex Graves. 2016. Memory-efficient backpropagation through time. In *Advances in Neural Information Processing Systems*, pages 4125–4133.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

|  | No Dropout |  | Dropout |  |
|---|---|---|---|---|
| Layers | dev(ppl) | News2017 | dev(ppl) | News2017 |
| 12 | 3.5 | 30.0 | 3.45 | 29.7 |
| 16 | 3.6 | 29.7 | 3.41 | 29.8 |

Table 4: Experiments using dropout, with large data including filtered paraCrawl.

| Layer | News20(13-16)<br>(ppl) | News2017<br>(BLEU) |
|---|---|---|
| 4 | 4.0 | 28.5 |
| 8 | 3.7 | 29.2 |
| 12 | 3.5 | 30.0 |
| 16 | 3.4 | 29.8 |
| 32 | **3.2** | **30.4** |

Table 5: Experiments using different layers, with large data including filtered paraCrawl.

| Layer | News2017 |
|---|---|
| 12-layer | 30.0 |
| 12-layer dropout | 29.7 |
| 16-layer | 29.7 |
| 16-layer dropout | 29.8 |
| 32-layer | 30.4 |
| ensemble | **30.7** |

Table 6: Systems used in the final submission

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

N-Q Pham. 2017. The karlsruhe institute of technology systems for the news translation task in wmt 2017. In *Proceedings of the Second Conference on Statistical Machine Translation (WMT 2017)*, Copenhagen, Dennmark.

R. Sennrich, B. Haddow, and A. Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54st Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

R. Sennrich, B. Haddow, and A. Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54st Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112, Quebec, Canada.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

# Tilde's Machine Translation Systems for WMT 2018

**Mārcis Pinnis** and **Matīss Rikters** and **Rihards Krišlauks**
Tilde / Vienibas gatve 75A, Riga, Latvia
{firstname.lastname}@tilde.lv

## Abstract

The paper describes the development process of the Tilde's NMT systems that were submitted for the WMT 2018 shared task on news translation. We describe the data filtering and pre-processing workflows, the NMT system training architectures, and automatic evaluation results. For the WMT 2018 shared task, we submitted seven systems (both constrained and unconstrained) for English-Estonian and Estonian-English translation directions. The submitted systems were trained using Transformer models.

## 1 Introduction

Neural machine translation (NMT) is a rapidly changing research area. Since 2016 when NMT systems first showed to achieve significantly better results than statistical machine translation (SMT) systems (Bojar et al., 2016), the dominant neural network (NN) architectures for NMT have changed on a yearly (and even more frequent) basis. The state-of-the-art in 2016 were shallow attention-based recurrent neural networks (RNN) with gated recurrent units (GRU) (Sennrich et al., 2016) in recurrent layers. In 2017 (Bojar et al., 2017), multiplicative long short-term memory (MLSTM) units (Pinnis et al., 2017c) and deep GRU (Sennrich et al., 2017a) models were introduced in NMT. The same year, self-attentional (Transformer) models were introduced (Vaswani et al., 2017). Consequently, in 2018, most of the top scoring systems in the shared task on news translation of the Third Conference on Machine Translation (WMT) were trained using Transformer models[1]. However, it is already evident that the state-of-the-art architectures will be pushed even further in 2018 (beyond WMT 2018). For instance, Chen et al. (2018) have recently proposed RNMT+ models that combine deep LSTM-based models with multi-head attention and showed that the models outperform Transformer models.

In WMT 2017, Tilde participated with MLSTM-based NMT systems (Pinnis et al., 2017c). In this paper, we compare the MLSTM-based models with Transformer models for English-Estonian and Estonian-English and we show that the state-of-the-art of WMT 2017 is well behind the new models. Therefore, for WMT 2018, Tilde submitted NMT systems that were trained using Transformer models.

The paper is further structured as follows: Section 2 provides an overview of systems submitted for the WMT 2018 shared task on news translation, Section 3 describes the data used to train the NMT systems and the data pre-processing workflows, Section 4 describes all NMT systems trained and experiments on handling of named entities and combination of systems, Section 5 provides automatic evaluation results, and Section 6 concludes the paper.

## 2 System Overview

For the WMT 2018 shared task on news translation, Tilde submitted both constrained and unconstrained NMT systems (7 in total). The following is a list of the five MT systems submitted:

- Constrained English-Estonian and Estonian-English NMT systems (*tilde-c-nmt*) that were deployed as ensembles of averaged factored data (see Section 3) Transformer models. The models were trained using parallel data and back-translated data in a 1-to-1 proportion.
- Unconstrained English-Estonian and Estonian-English NMT systems (*tilde-*

[1] All 14 of the best automatically scored systems according to the information provided by participants in the official submission portal http://matrix.statmt.org were indicated as being based on Transformer models.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 473–481
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64050

*nc-nmt*) that were deployed as averaged Transformer models. These models were also trained using back-translated data similarly to the constrained systems, however, the data, taking into account their relatively large size, were not factored.

- A constrained Estonian-English NMT system (*tilde-c-nmt-comb*) that is a system combination of six factored data NMT systems.

- Constrained English-Estonian and Estonian-English NMT systems (*tilde-c-nmt-2bt*) averaged from multiple best NMT models. The models were trained using two sets of back-translated data in a 1-to-1 proportion to the clean parallel data – one set was back-translated using a system trained on parallel-only data and the other set – using an NMT system trained on parallel data and the first set of back-translated data.

## 3 Data

Data preparation was done using one of two distinct workflows – we used the full workflow for *tilde-c-nmt*, *tilde-nc-nmt* and *tilde-c-nmt-comb* submissions. For the *tilde-c-nmt-2bt* submission we used the light data preparation workflow.

### 3.1 Full Workflow

For training of the constrained systems, only data provided by the WMT 2018 organisers were used, however, for training of the unconstrained systems, we also used other publicly available and proprietary corpora that were available in the Tilde Data Library[2]. All parallel corpora were filtered (see Section 3.1.1), pre-processed (see Section 3.1.2), and supplemented with additional generated data (see Section 3.1.3).

#### 3.1.1 Data Filtering

As NMT systems are sensitive to noise in parallel data (Pinnis et al., 2017a), all parallel data were filtered using the parallel data filtering methods described by Pinnis (2018). The parallel corpora filtering methods remove sentence pairs that have indications of data corruption or low parallelity (e.g., source-target length ratio, content overlap, digit mismatch, language adherence, etc.) issues. Contrary to Tilde's submissions for WMT 2017, isolated sentence pair filtering for the WMT 2018 submissions was supplemented with a maximum content overlap filter (i.e. only one target sentence for each source sentence was preserved and vice versa based on the content overlap filter's score for each sentence pair).

For filtering, we required probabilistic dictionaries, which were obtained from the parallel corpora (different dictionaries for the constrained and unconstrained scenarios) using *fast_align* (Dyer et al., 2013). The dictionaries were filtered using the transliteration-based probabilistic dictionary filtering method by Aker et al. (2014).

During filtering, we identified that one of the corpora that were provided by the organisers contained a significant amount of data corruption. It was the Estonian↔English ParaCrawl corpus[3]. The corpus consisted of 1.30 million sentence pairs out of which 0.77 million were identified as being corrupt. To reduce the high level of noise, this corpus was filtered using stricter content overlap (a threshold of 0.3 instead of 0.1) and language adherence filters (both the language detection and the valid alphabet filters had to validate a sentence pair instead of just one of the filters) than all other corpora. As a result, only 0.17 million sentence pairs from the ParaCrawl corpus were used for training of the constrained systems. Due to the quality concerns, the corpus was not used for training of the unconstrained systems.

The corpora statistics before and after filtering are provided in Table 1.

#### 3.1.2 Data Pre-processing

All corpora were pre-processed using the parallel data pre-processing workflow from the Tilde MT platform (Pinnis et al., 2018) that performs the following pre-processing steps:

- First, parallel corpora are cleaned by removing HTML and XML tags, decoding escaped symbols, normalising whitespaces and punctuation marks, replacing control characters with spaces, etc. This step is performed only on the training data.

- Then, non-translatable entities, such as e-mail addresses, URLs, file paths, etc. are identified and replaced with place-holders. This allows reducing data sparsity where it is not needed.

[2] Tilde Data Library is an integral component of the Tilde MT platform that provides access to parallel and monolingual data for MT system development (`http://www.tilde.com/mt/`).

[3] `https://paracrawl.eu/download.html`

| Workflow | Scenario | Before filtering (Total / Unique) | After filtering (Unique) |
|---|---|---|---|
| Full | (C) | 2,178,025 / 1,932,954 | 968,232 |
| | (U) | 75,215,347 / 24,660,087 | 18,755,230 |
| Light | (C) | 2,178,025 | 998,679 |

Table 1: Training data statistics (sentence counts) before and after filtering

- Then, the data are tokenised using the Tilde MT regular expression-based tokeniser.
- The Moses (Koehn et al., 2007) truecasing script *truecase.perl* is used to truecase the first word of every sentence.
- Then, tokens are split into sub-word units (Sennrich et al., 2015) using byte-pair encoding (BPE) (Gage, 1994). For the constrained and unconstrained systems, we use BPE models consisting of 24,500 and 49,500 merging operations respectively.
- Finally, data for the constrained systems are factored using an averaged perceptron-based morpho-syntactic tagger (Nikiforovs, 2014) for Estonian and the lexicalized probabilistic parser (Klein et al., 2002) from the *Stanford CoreNLP* toolkit (Manning et al., 2014) for English. Similarly to Sennrich and Haddow (2016), we introduce also a factor indicating a word part's position in a word (beginning, middle, end, or the word part represents the whole word - *B*, *I*, *E*, or *O*). As a result, the Estonian data consist of the the following factors: *word part*, *position*, *lemma*, and *morpho-syntactic tag*. The English data consist of the following factors: *word part*, *position*, *lemma*, *part-of-speech tag*, and *syntactic function*.

### 3.1.3 Synthetic Data

Similarly to Tilde's 2017 systems (Pinnis et al., 2017c), we submitted systems that were trained using synthetic data: 1) back-translated data, and 2) data infused with unknown token identifiers. The back-translated data allow performing domain adaptation and the second type of synthetic data allow training NMT models that are robust to unknown phenomena (e.g., code-mixed content, target language words in the source text, rare or unseen words, etc.) (Pinnis et al., 2017b).

To create the synthetic corpora with unknown phenomena, we extracted *fast_align* (Dyer et al., 2013) word alignments for each sentence pair in the parallel corpora and randomly replaced one to three unambiguously (one-to-one) aligned content words with unknown word identifiers. These synthetic corpora were added to the parallel corpora, thereby almost doubling the sizes of the available training data.

|  | Lang. pair | Back-transl. sent. | Synth. <*UNK*> sent. | Total |
|---|---|---|---|---|
| ***Full workflow*** | | | | |
| (C) | en-et | 0.97M | 1.72M | 3.65M |
| | et-en | 0.97M | 1.79M | 3.73M |
| (U) | en-et | 16.21M | 28.10M | 63.07M |
| | et-en | 18.39M | 30.77M | 67.91M |
| ***Light workflow*** | | | | |
| (C) | en-et | 2.11M | | 3.11M |
| | et-en | 2.05M | | 3.04M |

Table 2: Synthetic data and final NMT model training data statistics

The back-translated data were acquired from two sources: 1) the constrained system data were acquired from initial Transformer-based NMT systems that were trained on the filtered and pre-processed parallel data, which were supplemented with the unknown phenomena infused data, and 2) the unconstrained system data were acquired from pre-existing unconstrained MLSTM-based NMT systems – the NMT systems that were developed by Tilde for the Estonian EU Council Presidency in 2017 (Pinnis and Kalniņš, 2018). In order to limit noise, the back-translated data were filtered using the same parallel data filtering methods that were described in Section 3.1.1 (although with a higher threshold for the content overlap filter). Furthermore, in order to train the final systems, we also generated unknown phenomena infused data for the back-translated filtered data, thereby also almost doubling the sizes of the back-translated data.

The synthetic corpora statistics and the sizes of the total training data are given in Table 2.

|  | **Name** | **Model** | **Voc.** | **Emb. layer (f1:..:fN)** | **Other layers (enc:dec, size)** | **Seq. len.** |
|---|---|---|---|---|---|---|
| ***English-Estonian*** |  |  |  |  |  |  |
| (C) | MLSTM | MLSTM | 25k | 350:5:125:10:10 | 1:1 1024 | 80 |
|  | transf | Transformer |  | 512:5:125:11:11 | 6:6, model: 512 | 128 |
|  | transf-2bt |  | 50k | 512 |  |  |
|  | transf-l |  | 25k | 720:5:125:11:11 | 7:7, model: 720 |  |
| (U) | transf-u |  | 50k | 512 |  |  |
| ***Estonian-English*** |  |  |  |  |  |  |
| (C) | MLSTM | MLSTM | 25k | 360:5:125:10 | 1:1 1024 | 80 |
|  | transf | Transformer |  | 512:5:125:14 | 6:6, model: 512 | 128 |
|  | transf-l |  |  | 720:5:125:14 | 7:7, model: 720 |  |
|  | transf-l2 |  |  | 1024:5:125:14 | 8:8, model: 1024 |  |
|  | transf-2bt |  | 50k | 512 | 6:6, model: 512 |  |
| (U) | transf-u |  |  | 720 |  |  |

Table 3: NMT system training configuration (all other parameters were set to the default values of the respective toolkits (Nematus or Sockeye)

### 3.2 Light Workflow

In the light workflow we used data cleaning and pre-processing methods described by Rikters (2018). The filtering part includes the following filters: 1) unique parallel sentence filter; 2) equal source-target filter; 3) multiple sources - one target and multiple targets - one source filters; 4) non-alphabetical filters; 5) repeating token filter; and 6) correct language filter. The pre-processing consists of the standard Moses (Koehn et al., 2007) scripts for tokenising, cleaning, truecasing, and Subword NMT for splitting into subword units. The filters were applied to the given parallel sentences, monolingual news sentences before performing back-translation, and both sets of synthetic parallel sentences that resulted from back-translating the monolingual news.

## 4 NMT Systems

In order to train the NMT systems, we used the Nematus (Sennrich et al., 2017b) (for MLSTM models) and Sockeye (Hieber et al., 2017) (for Transformer models) toolkits. All models were trained until convergence (i.e., until an early stopping criterion was met).

### 4.1 Full Workflow

First, we trained constrained system baseline models using the filtered datasets. For baseline models, we used the *MLSTM* and *transf* configurations (see Table 3). Then, we used the best-performing models (based on translation quality on the validation set), which were the Transformer models (see Figure 1), and back-translated monolingual data. As mentioned before, for the unconstrained systems, we back-translated the monolingual data using pre-existing MLSTM-based NMT systems. Then, using the final training data (parallel and the two synthetic corpora), we trained final Transformer models. For the constrained scenario, we trained multiple models (three for each translation direction) by experimenting with multiple model configurations. For the unconstrained scenario, we trained one model in each of the directions.

In order to acquire the translations for the submissions, we performed model averaging and ensembling as follows:

- For the *tilde-c-nmt* (constrained NMT) systems, we performed model averaging of the best four models (according to perplexity) of the three different run NMT systems and deployed the averaged models in an ensemble.
- For the *tilde-nc-nmt* (unconstrained NMT) systems, we performed model averaging of the best four models.
- For the *tilde-c-nmt-comb* Estonian-English system, we performed majority voting (see Section 4.3) of translations produced by six different runs of different constrained systems (using best BLEU (Papineni et al., 2002) models, averaged models, ensembled averaged models, ensembled models, and larger beam search (10 instead of 5)).

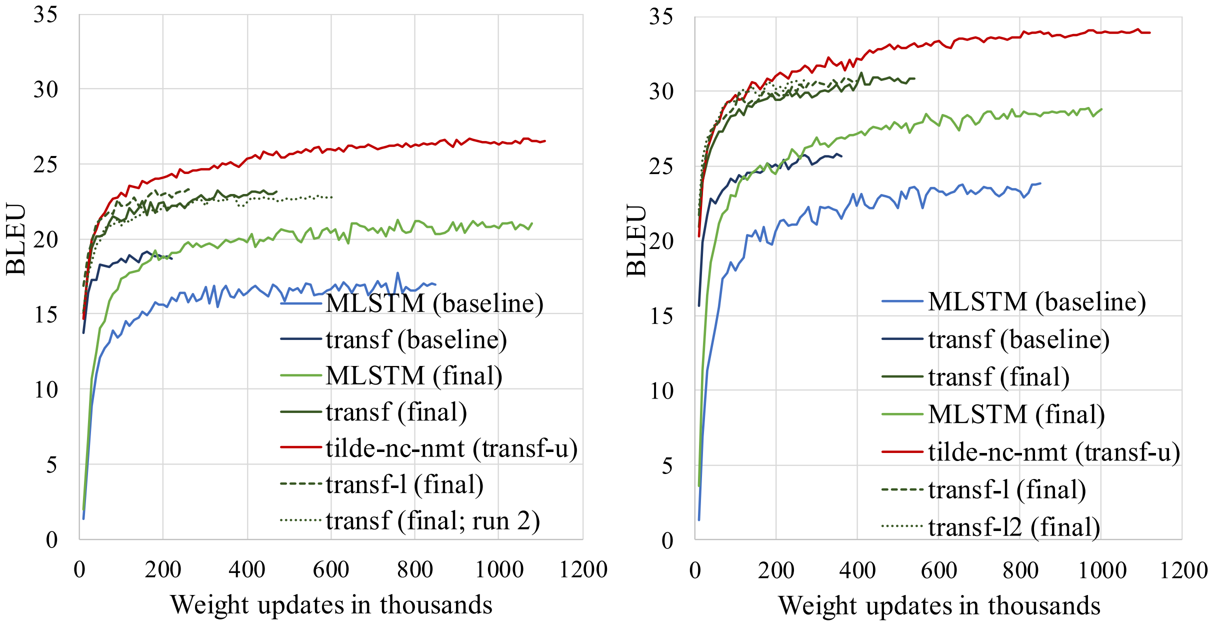


Figure 1: NMT system training progress (BLEU scores on the validation set) for English-Estonian (left) and Estonian-English (right). Note that batch size may differ between different architectures and BLEU scores are calculated on raw (token level) pre-processed validation sets, therefore, the scores are slightly higher than evaluation results for the final translations!

### 4.1.1 Automatic Post-editing of Named Entities

NMT models so far have struggled with translating rare or unseen words (not different surface forms, but rather different words) correctly (Pinnis et al., 2017c). Named entities and non-translatable entities (various product names, identifiers, etc.) are often rare or unknown. In order to aid the NMT model in translating such tokens better, we extracted named entity and non-translatable token dictionaries from the parallel corpora. This was done by performing word alignment of the parallel corpora using *fast_align* (Dyer et al., 2013) and searching (in a language-agnostic manner) for transliterated source-target word pairs using a similarity metric based on Levenshtein distance (Levenshtein, 1966), which start with upper-case letters. The dictionaries consist of 15.6 (94.7) thousand and 6.2 (149.8) thousand entries for the constrained (unconstrained) English-Estonian and Estonian-English NMT systems respectively.

When the NMT systems had translated a sentence, source-to-target word alignment was extracted from the source sentence and the translation. Then named entity recognition (based on dictionary look-up) was performed on the source text and, if a named entity was found, the target translation was validated against the entries in the dictionary. In order to capture different surface forms, a stemming tool was used. If a translation was contradicting the entries in the dictionary, it was replaced with the closest matching (by looking for the longest matching suffix) translation from the dictionary.

The automatic post-editing method for named entities has a marginal impact on translation quality, however, manual analysis showed that more named entities were corrected than ruined.

## 4.2 Light Workflow

The light workflow was used to produce the *tilde-c-nmt-2bt* (constrained NMT with two sets of back-translated data) systems. First, we trained baseline models using only filtered parallel datasets (Parallel-only in Figure 2). Then, we back-translated the first batches of monolingual news data and trained intermediate NMT systems (Parallel + First Back-translated). Finally, we used the intermediate NMT systems to back-translate the second batches of monolingual news data and trained final NMT systems (Parallel + Second Back-translated). The training progress in Figure 2 shows that the English-Estonian system benefits from the additional data, but the system in the other direction – not so much.

For the final translations, we used a post-processing script (Rikters et al., 2017) to replace

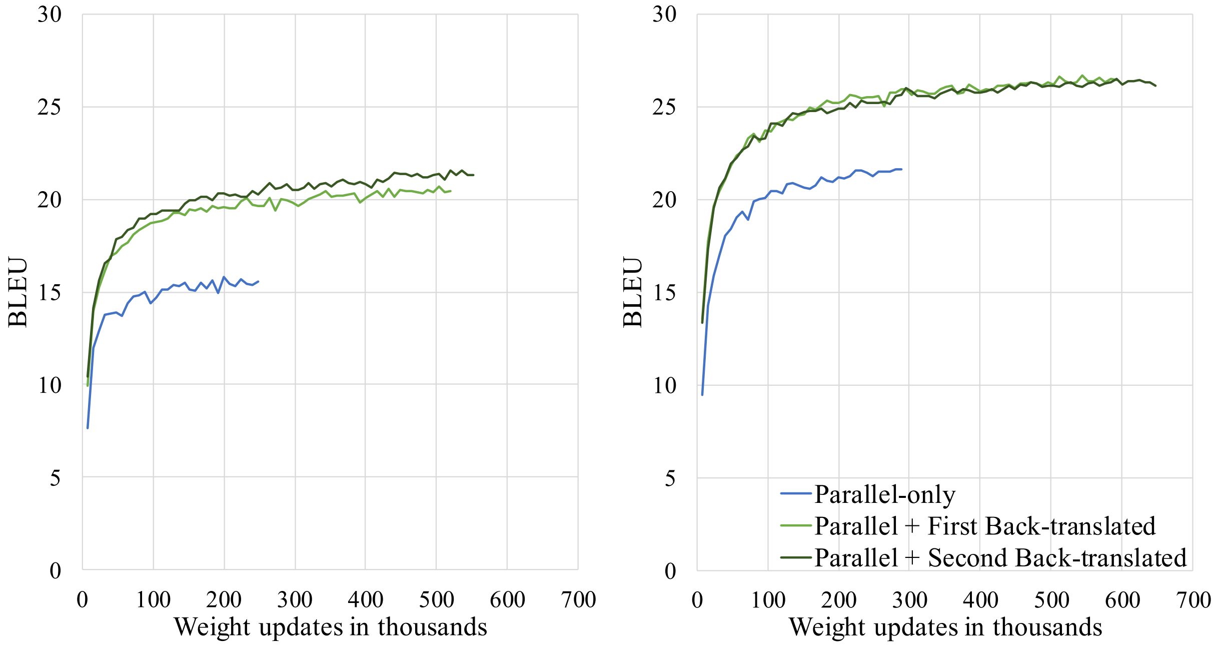


Figure 2: NMT system training progress (SacreBLEU scores on the validation set) for English-Estonian (left) and Estonian-English (right).

consecutive repeating n-grams and repeating n-grams that have a preposition between them (i.e., *victim of the victim*) with a single n-gram. This problem was more apparent in RNN-based NMT systems, but it was also noticable in our Transformer model outputs.

### 4.3 System Combination

We attempted to increase the quality of existing translations by employing a voting scheme in which multiple machine translation outputs are combined to produce a single translation. We used a custom implementation of the majority voting algorithm (Freitag et al., 2014) to combine six of our best-scoring outputs in the Estonian-English translation direction in the constrained scenario. We did not perform the combination for English-Estonian due to lack of support for alignment extraction for Estonian in Meteor (Denkowski and Lavie, 2014).

MT system translation combination happens on the sentence level. The majority voting scheme assumes a single base translation hypothesis (primary hypothesis) which is aligned at the word level to each of the other hypotheses (secondary hypotheses). The alignments are used to generate a table of all possible word translations relative to each position in the primary hypothesis. The table is then used to count the number of occurrences of different translations. The word translations with the highest count at each position constitute the resulting combined hypothesis.

To acquire the necessary word alignments we used Meteor. Meteor outputs were then converted to a more easily manageable form using the Jane toolkit (Freitag et al., 2014) (we used an *awk* script distributed with Jane). The majority voting algorithm was implemented in *Python*.

## 5 Results

We performed automatic evaluation of the NMT systems using the SacreBLEU evaluation tool (Post, 2018). The results (see Table 4) show that the Transformer models achieved better results than the MLSTM-based models. For the constrained scenarios, both ensembles of averaged models achieved higher scores than each individual averaged model. It is also evident that the unconstrained models (*tilde-nc-nmt*) achieved the best results.

Although the unconstrained models were not trained on factored data, the datasets were 17 times larger than the constrained datasets. However, the difference is rather minimal and shows that the current NMT architectures may not able to learn effectively from large datasets.

The official human evaluation results (see Table 5) from the WMT 2018 shared task on news translation (Bojar et al., 2018) show that

| System | Configuration | BLEU |
|---|---|---|
| ***English-Estonian*** | | |
| MLSTM (final) | 5 model ensemble | 20.80 |
| transf (final) | 4 model average | 22.82 |
| transf-l (final) | | 23.04 |
| transf (final; run 2) | | 22.56 |
| tilde-c-nmt | ensemble of 3 averaged models | **23.54** |
| tilde-c-nmt-2bt | 3 model average | **23.57** |
| tilde-nc-nmt (transf-u) | 4 model average | **24.35** |
| ***Estonian-English*** | | |
| MLSTM (final) | 5 model ensemble | 26.79 |
| transf (final) | 4 model average | 28.14 |
| transf-l (final) | | 28.83 |
| transf-l2 (final) | | 25.40 |
| tilde-c-nmt | ensemble of 3 averaged models | **29.46** |
| tilde-c-nmt-comb | 6 system combination | **29.36** |
| tilde-c-nmt-2bt | 3 model average | 27.99 |
| tilde-nc-nmt (transf-u) | 4 model average | **30.94** |

Table 4: Automatic evaluation results

| | System | BLEU | DA | Cluster |
|---|---|---|---|---|
| ***English-Estonian*** | | | | |
| | nict | 25.16 | 62.1 | 2 |
| (C) | tilde-c-nmt | 23.54 | 61.6 | 2 |
| | aalto | 20.66 | 58.6 | 5 |
| | tilde-nc-nmt | 24.35 | 64.9 | 1 |
| (U) | online-b | 18.71 | 52.1 | 10 |
| | neurotolge.ee | 15.53 | 45.7 | 11 |
| ***Estonian-English*** | | | | |
| | nict | 30.68 | 71.1 | 2 |
| (C) | tilde-c-nmt | 29.46 | 69.9 | 2 |
| | uedin | 29.38 | 69.2 | 2 |
| | tilde-nc-nmt | 30.94 | 73.3 | 1 |
| (U) | online-b | 25.81 | 67.1 | 2 |
| | online-a | 22.44 | 65.4 | 10 |

Table 5: Top three systems for the constrained (C) and unconstrained (U) scenarios according to the official results of the WMT 2018 shared task on news translation; ordered by the direct assessment (DA) standardized mean score

our unconstrained scenario systems (*tilde-nc-nmt*) ranked significantly higher than any other submission for both translation directions. Our best constrained systems were the second highest ranked systems among all constrained scenario systems, at the same time sharing the same cluster with the highest ranked systems.

## 6 Conclusion

The paper described the development process of the Tilde's NMT systems that were submitted for the WMT 2018 shared task on news translation. We compared Transformer models to MLSTM-based models and showed that the Transformer models outperform the older NMT architecture. We also showed that double back-translation may improve translation quality further than single back-translation. In terms of model ensembling and averaging, we showed that the best results in the constrained scenario were achieved by ensembling different run averaged models. In total, seven systems were submitted by Tilde for the English↔Estonian language pair.

## Acknowledgements

The research has been supported by the European Regional Development Fund within the research project "Neural Network Modelling for Inflected Natural Languages" No. 1.1.1.1/16/A/215.

## References

Ahmet Aker, Monica Lestari Paramita, Mārcis Pinnis, and Robert Gaizauskas. 2014. Bilingual Dictionaries for All EU Languages. In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC'14)*, pages 2839–2845, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013)*, June, pages 644–648, Atlanta, USA.

Markus Freitag, Matthias Huck, and Hermann Ney. 2014. Jane: Open source machine translation system combination. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *ArXiv e-prints*.

Dan Klein, Christopher D Manning, et al. 2002. Fast exact inference with a factored model for natural language parsing. *Advances in Neural Information Processing Systems (NIPS 2002)*, pages 3–10.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vladimir I Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10(8):707–710.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the ACL 2014 System Demonstrations*, pages 55–60.

Peteris Nikiforovs. 2014. Latvian NLP: Perceptron Tagger.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Mārcis Pinnis. 2018. Tilde's Parallel Corpus Filtering Methods for WMT 2018. In *Proceedings of the Third Conference on Machine Translation (WMT 2018), Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Mārcis Pinnis and Rihards Kalniņš. 2018. Developing a Neural Machine Translation Service for the 2017-2018 European Union Presidency. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (AMTA 2018), vol. 2: MT Users*, pages 72–83, Boston, USA. Association for Machine Translation in the Americas.

Mārcis Pinnis, Rihards Krišlauks, Daiga Deksne, and Toms Miks. 2017a. Evaluation of Neural Machine Translation for Highly Inflected and Small Languages. In *Proceedings of the 18th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2017)*, Budapest, Hungary.

Mārcis Pinnis, Rihards Krišlauks, Daiga Deksne, and Toms Miks. 2017b. Neural Machine Translation for Morphologically Rich Languages with Improved Sub-word Units and Synthetic Data. In *Proceedings of the 20th International Conference of Text, Speech and Dialogue (TSD2017)*, volume 10415 LNAI, Prague, Czechia.

Mārcis Pinnis, Rihards Krišlauks, Toms Miks, Daiga Deksne, and Valters Šics. 2017c. Tilde's Machine Translation Systems for WMT 2017. In *Proceedings of the Second Conference on Machine Translation (WMT 2017), Volume 2: Shared Task Papers*, pages 374–381, Copenhagen, Denmark. Association for Computational Linguistics.

Mārcis Pinnis, Andrejs Vasiļjevs, Rihards Kalniņš, Roberts Rozis, Raivis Skadiņš, and Valters Šics. 2018. Tilde MT Platform for Developing Client Specific MT Solutions. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Matīss Rikters. 2018. Impact of Corpora Quality on Neural Machine Translation. In *In Proceedings of the 8th Conference Human Language Technologies - The Baltic Perspective (Baltic HLT 2018)*, Tartu, Estonia.

Matīss Rikters, Chantal Amrhein, Maksym Del, and Mark Fishel. 2017. C-3MA: Tartu-Riga-Zurich Translation Systems for WMT17. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017a. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Others. 2017b. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68.

Rico Sennrich and Barry Haddow. 2016. Linguistic Input Features Improve Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation (WMT 2016) - Volume 1: Research Papers*, pages 83–91.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

# CUNI Transformer Neural MT System for WMT18

**Martin Popel**
Charles University, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics,
Prague, Czechia
popel@ufal.mff.cuni.cz

## Abstract

We describe our NMT system submitted to the WMT2018 shared task in news translation. Our system is based on the Transformer model (Vaswani et al., 2017). We use an improved technique of backtranslation, where we iterate the process of translating monolingual data in one direction and training an NMT model for the opposite direction using synthetic parallel data. We apply a simple but effective filtering of the synthetic data. We pre-process the input sentences using coreference resolution in order to disambiguate the gender of pro-dropped personal pronouns. Finally, we apply two simple post-processing substitutions on the translated output.

Our system is significantly ($p < 0.05$) better than all other English-Czech and Czech-English systems in WMT2018.

## 1 Introduction

The quality of Neural Machine Translation (NMT) depends heavily on the amount and quality of the training parallel sentences as well as on various training tricks, which are sometimes surprisingly simple and effective.

In this paper, we describe our NMT system "CUNI Transformer" (Charles University version of Transformer), submitted to the English→Czech and Czech→English news translation shared task within WMT2018. We describe five techniques, which helped to improve our system, so that it outperformed all other systems in these two translation directions: training data filtering (Section 3), improved backtranslation (Section 4), tuning two separate models based on the original language of the text to be translated (Section 5), coreference pre-processing (Section 6) and post-processing using regular expressions (Section 7). Our system significantly outperformed all other systems in WMT2018 evaluation (Section 8).

| data set | sentence pairs (k) | words (k) EN | words (k) CS |
|---|---:|---:|---:|
| CzEng 1.7 | 57 065 | 618 424 | 543 184 |
| Europarl v7 | 647 | 15 625 | 13 000 |
| News Commentary v12 | 211 | 4 544 | 4 057 |
| CommonCrawl | 162 | 3 349 | 2 927 |
| EN NewsCrawl 2016–17 | 47 483 | 934 981 | |
| CS NewsCrawl 2007–17 | 65 383 | | 927 348 |
| total | 170 951 | 1 576 923 | 1 490 516 |

Table 1: Training data sizes (in thousands).

## 2 Experimental Setup

Our training data is constrained to the data allowed in the WMT2018 shared task. Parallel (authentic) data are: CzEng 1.7, Europarl v7, News Commentary v11 and CommonCrawl. In our backtranslation experiments (Section 4), we used synthetic data translated by backtranslation of monolingual data: Czech and (a subset of) English NewsCrawl articles. We filtered out ca. 3% of sentences from the synthetic data (Section 3). Data sizes are reported in Table 1.

Note that usually the amount of available monolingual data is orders of magnitude larger than the available parallel data, but in our case it is comparable (58M parallel vs. 65M/48M monolingual). We used all the Czech monolingual data allowed in the constrained task.

We used the Transformer self-attentional sequence-to-sequence model (Vaswani et al., 2017) implemented in the Tensor2Tensor framework.[1] We followed the training setup and tips of Popel and Bojar (2018), but we trained our models with the Adafactor optimizer (Shazeer and Stern, 2018) instead of the default Adam: We used T2T version 1.6.0, `transformer_big` and hyper-parameters `learning_rate_schedule=rsqrt_decay`,

[1] https://github.com/tensorflow/tensor2tensor

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 482–487
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64051

`learning_rate_warmup_steps=8000`, `batch_size=2900`, `max_length=150`, `layer_prepostprocess_dropout=0`, `optimizer=Adafactor`. For decoding, we used `alpha=1`.

We stored model checkpoints each hour and averaged the last eight checkpoints. We used eight GTX 1080 Ti GPUs.

## 3 Training Data Filtering

We found out that the Czech monolingual data set (NewsCrawl 2007–2017) contains many English sentences. Those sentences were either kept untranslated or paraphrased when preparing the synthetic data with backtranslation. Thus the synthetic data included many English-English sentence pairs. Consequently, the synth-trained models had a higher probability of keeping a sentence untranslated.

In order to filter out the English sentences from the Czech data, we kept only sentences containing at least one accented character.[2] We also filtered out sentences longer than 500 characters from the synthetic data. Most of these sentences would be ignored anyway because we are training our Transformer with `max_length`=150, i.e. filtering out sentences longer than 150 subwords (cf. Popel and Bojar, 2018, § 4.4). Sometimes a Czech sentence was much shorter than its English translation (especially for the translations by Nematus2016) – because of filler words repeated many times, which is a well-known problem of NMT systems (e.g. Sudarikov et al., 2016). We filtered out all sentences with a word (or a pair of words) repeated more than twice using a regular expression `/ (\S+ ?\S+) \1 \1 /`. This way, we filtered out ca. 3% of sentences and re-trained our systems. After this filtering, we did not observe any untranslated sentences in the synth-trained output.

## 4 Improved Backtranslation

Sennrich et al. (2016b) introduced backtranslation as a simple way how to utilize target-language monolingual data in NMT. The monolingual data sets are translated (by a target-to-source MT system) to the source language, resulting in *synthetic* parallel data, which is used as additional training data (in addition to *authentic* parallel) for the final (source-to-target) NMT system.

Sennrich et al. (2017) compared two regimes of how to incorporate synthetic training data created using backtranslation of monolingual data. In the *fine-tuned* regime, a system is trained first on the authentic parallel data and then after several epochs it is trained on a 1:1 mix of authentic and synthetic data. In the *mixed* regime, the 1:1 mixed data is used from the beginning of training. In both cases, the 1:1 mix means shuffling the data randomly at the sentence level, possibly oversampling the smaller of the two data sources.

We used a third approach, termed *concat* regime, where the authentic and synthetic parallel data are simply concatenated (without shuffling). We observed that this regime leads to improvements in translation quality relative to both *mixed* and *fine-tuned* regimes, especially when checkpoint averaging is used.

For obtaining the final English→Czech system, we iterated the backtranslation process:

1. We downloaded the Nematus2016 models trained by Sennrich et al. (2016a) using *fine-tuned* backtranslation of English NewsCrawl 2015 articles, which were translated "*with an earlier NMT model trained on WMT15 data*" (Sennrich et al., 2016a). We used these Nematus2016 models to translate Czech NewsCrawl 2007–2017 articles to English.

2. We trained an English→Czech Transformer on this data (filtered as described in Section 3) using concat backtranslation with checkpoint averaging. We used this Transformer model to translate English NewsCrawl 2016–2017 articles into Czech.

3. We trained our Czech→English Transformer model (used for our WMT18 submission) on this data using concat backtranslation with averaging. We translated Czech NewsCrawl 2016–2017 articles into English using this system, producing a higher-quality synthetic data than in step 1 (but smaller because of lack of time and resources).

4. We trained our final English→Czech system

[2] `m/[ěščřžýáíéúůd'tʼň]/i` – this simple heuristics is surprisingly effective for Czech. In addition to English sentences, it filters out also *some* short Czech sentences, sentences in other languages (e.g. Chinese) and various "non-linguistic" content, such as lists of football or stock-market results.

on this data, again using concat backtranslation with averaging.

Each training (steps 2, 3 and 4) took eight days on eight GPUs. Translating the monolingual data with Nematus2016 (step 1) took about two weeks and with our Transformer models (steps 2 and 3) took about five days. The final model trained in step 4 is +0.83 BLEU better than the model trained in step 2 without data filtering, as measured on newstest2017 (cf. Table 2).

## 5 CZ/nonCZ Tuning

In WMT test sets since 2014, half of the sentences for a language pair X-EN originate from English news servers (e.g. bbc.com) and the other half from X-language news servers. All WMT test sets include the server name for each document in metadata, so we were able to split our test set (and dev set newstest2013) into two parts: `CZ` (for Czech-domain articles, i.e. documents with `docid` containing ".cz") and `nonCZ` (for non-Czech-domain articles). We noticed that when training on synthetic data, the model performs much better on the `CZ` test set than on the `nonCZ` test set. When trained on authentic data, it is the other way round. Intuitively, this makes sense: The target side of our synthetic data are original Czech sentences from Czech newspapers, similarly to the `CZ` test set. In our authentic data, over 90% of sentences were originally written in English about "non-Czech topics" and translated into Czech (by human translators), similarly to the `nonCZ` test set. There are two closely related phenomena: a question of domain (topics) in the training data and a question of so-called *translationese* effect, i.e. which side of the parallel training data (and test data) is the original and which is the translation.

Based on these observations, we prepared a `CZ`-tuned model and a `nonCZ`-tuned model. Both models were trained in the same way, they differ only in the number of training steps. For the `CZ`-tuned model, we selected a checkpoint with the best performance on `wmt13-CZ` (Czech-origin portion of newstest2013), which was at 774k steps. Similarly, for the `nonCZ`-tuned model, we selected the checkpoint with the best performance on `wmt13-nonCZ`, which was at 788k steps. Note that both the models were trained jointly in one experiment, just selecting checkpoints at two different moments.

## 6 Coreference Pre-processing

In Czech, as a pro-drop language, it is common to omit personal pronouns in subject positions. Usually, the information about gender and number of the subject is encoded in the verb inflection, but present-tense verbs have the same form for the feminine and masculine gender. For example, "*Není doma*" can mean either "*She is not home*" or "*He is not home*". When translating such sentences from Czech to English, we must use the context of neighboring sentences in a given document, in order to disambiguate the gender and select the correct translation. However, our Transformer system (similarly to most current NMT systems) translates each sentence independently of other sentences. We observed that in practice it always prefers the masculine gender if the information about gender could not be deduced from the source sentence.

We implemented a simple pre-processing of the Czech sentences, which are then translated with our Czech→English Transformer system – we inserted pronoun *ona* (*she*), where it was "missing". We analyzed the source Czech documents in the Treex NLP framework (Popel and Žabokrtský, 2010), which integrates a coreference resolver (Novák, 2017). We found sentences where a female-gender pronoun subject was dropped and the coreference link was pointing to a different sentence (usually the previous one). We restricted the insertion of *ona* only to the cases in which the antecedent in the coreference chain represents a human (i.e. excluding grammatical-only female gender of inanimate objects and animals). We used a heuristic detection of human entities, which is integrated in Treex.

This preprocessing affected only 1% of sentences in our nestest2017 dev set and for most of them the English translation was improved (according to our judgment), although the overall BLEU score remained the same. We consider this solution as a temporary workaround before document-level NMT (e.g. Kuang et al., 2017) is available in T2T. That said, the advantage of the described preprocessing is that it can be applied to any (N)MT system – without changing its architecture and even without retraining it.

## 7 RegEx Post-processing

We applied two simple post-processings to the translations, using regular expressions.

| English→Czech<br>system | BLEU<br>cased | BLEU<br>uncased | chrF2<br>cased |
|---|---|---|---|
| Nematus (Sennrich et al., 2016b) | 22.80 | 23.29 | 0.5059 |
| T2T (Popel and Bojar, 2018) | 23.84 | 24.40 | 0.5164 |
| our mixed backtranslation | 24.85 (+1.01) | 25.33 | 0.5267 |
| our concat backtranslation | 25.77 (+0.92) | 26.29 | 0.5352 |
| + higher quality backtranslation | 26.60 (+0.83) | 27.10 | 0.5410 |
| + CZ/nonCZ tuning | **26.81** (+0.21) | **27.30** | **0.5431** |

Table 2: Automatic evaluation on (English→Czech) `newstest2017`. The three scores in parenthesis show BLEU difference relative to the previous line.

We deleted phrases repeated more than twice (immediately following each other); we kept just the first occurrence. We considered phrases of one up to four words. With the training-data filtering described in Section 3, less than 1% sentences needed this post-processing.

For English→Czech, we converted quotation symbols in the translations to the correct-Czech „lower and upper“ quotes using two regexes: `s/(^|[ ({[])("|,,|''|``)/$1„/g` and `s/("|'')($|[ ,.?!:;)}\]])/“$2/g`. In English, the distinction between "straight" and “curly” quotes is considered as a rather typographical (or style-related) issue. However, in Czech, a mismatch between lower (opening) and upper (closing) quotes is considered as an error in formal writing.

# 8 Evaluation

## 8.1 WMT2017 Evaluation

Table 2 evaluates the relative improvements described in Sections 4 and 5 on English→Czech newstest2017 and compares the results with the WMT2017 winner – Nematus (Sennrich et al., 2016b), and with the result of Popel and Bojar (2018) – T2T without any backtranslation.

The three reported automatic metrics are: case-sensitive (cased) BLEU, case-insensitive (uncased) BLEU and a character-level metric chrF2 (Popović, 2015). We compute all the three metrics with sacreBLEU (Post, 2018). The reported cased and uncased variants of BLEU differ also in the tokenization. The *cased* variant uses the default (ASCII-only) for better comparability with the results at `http://matrix.statmt.org`. The *uncased* variant uses the international tokenization, which has higher correlation with humans (Macháček and Bojar, 2013). The sacreBLEU signatures of the three metrics are:

- `BLEU+case.mixed+lang.en-cs+numrefs.1+smooth.exp+test.wmt17+tok.13a`,
- `BLEU+case.lc+lang.en-cs+numrefs.1+smooth.exp+test.wmt17+tok.intl` and
- `chrF2+case.mixed+lang.en-cs+numchars.6+numrefs.1+space.False+test.wmt17`.

We performed a small-scale manual evaluation on newstest2017 and noticed that in many cases the human reference translation is actually worse than our Transformer output. Thus the results of BLEU (or any other automatic metric comparing similarity with references) may be misleading.

## 8.2 WMT2018 Evaluation

Table 3 the reports results of all English↔Czech systems submitted to WMT2018, according to both automatic and manual evaluation. For the automatic evaluation, we use the same three metrics as in the previous section (just with `wmt18` instead of `wmt17`). For the manual evaluation, we report the reference-based direct assessment (refDA) scores, provided by the WMT organizers.

Our Transformer is the best system in English→Czech and Czech→English WMT2018 news task. It is significantly ($p < 0.05$) better than the second-best system – UEdin NMT, in both translation directions and both according to BLEU bootstrap resampling test (Koehn, 2004) and according to refDA Wilcoxon rank-sum test.

# 9 Conclusion

We have presented five simple but effective techniques for improving (N)MT quality. All five tech-

|  | English→Czech |  |  |  | Czech→English |  |  |  |
|---|---|---|---|---|---|---|---|---|
| system | BLEU uncased | BLEU cased | chrF2 cased | refDA Ave. % | BLEU uncased | BLEU cased | chrF2 cased | refDA Ave. % |
| our Transformer | **26.82** | **26.01** | **0.5372** | **67.2** | **35.64** | **33.91** | **0.5876** | **71.8** |
| UEdin NMT | 24.30 | 23.42 | 0.5166 | 60.6 | 34.12 | 33.06 | 0.5801 | 67.9 |
| Online-B | 20.16 | 19.45 | 0.4854 | 52.1 | 33.58 | 31.78 | 0.5736 | 66.6 |
| Online-A | 16.84 | 15.74 | 0.4584 | 46.0 | 28.47 | 26.78 | 0.5447 | 62.1 |
| Online-G | 16.33 | 15.11 | 0.4560 | 42.0 | 25.20 | 22.53 | 0.5310 | 57.5 |

Table 3: WMT2018 automatic (BLEU, chrF2) and manual (refDA = reference-based direct assessment) evaluation on `newstest2018`.

niques can be applied to virtually any NMT system. According to the preliminary results of the manual evaluation, the final translation quality is comparable to or even better than the quality of human references.

As a future work, we would like to assess the relative improvement of each of the five techniques based on manual evaluation (because automatic single-reference evaluation is not reliable when the MT quality is near to the quality of reference translations).

## Acknowledgements

The work described in this paper has been supported by the "NAKI II - Systém pro trvalé uchování dokumentace a prezentaci historichých pramenů z období totalitních režimů", project No. DG16P02B048, funded by the Ministry of Culture of the Czech Republic.

## References

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, volume 4, pages 388–395.

Shaohui Kuang, Deyi Xiong, Weihua Luo, and Guodong Zhou. 2017. Cache-based Document-level Neural Machine Translation. *CoRR*, arXiv/1711.11221.

Matouš Macháček and Ondřej Bojar. 2013. Results of the WMT13 Metrics Shared Task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 45–51, Sofia, Bulgaria. Association for Computational Linguistics.

Michal Novák. 2017. Coreference resolution system not only for czech. In *Proceedings of the 17th conference ITAT 2017: Slovenskočeský NLP workshop (SloNLP 2017)*, pages 193–200, Praha, Czechia. CreateSpace Independent Publishing Platform.

Martin Popel and Ondřej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110:43–70.

Martin Popel and Zdeněk Žabokrtský. 2010. TectoMT: Modular NLP Framework. *Advances in Natural Language Processing*, pages 293–304.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. ACL.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. *CoRR*, arXiv/1804.08771.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's Neural MT Systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. *CoRR*, arXiv/1804.04235.

Roman Sudarikov, Martin Popel, Ondřej Bojar, Aljoscha Burchardt, and Ondřej Klejch. 2016. Using MT-ComparEval. In *Translation Evaluation:*

*From Fragmented Tools and Data Sets to an Integrated Ecosystem*, pages 76–82, Portorož, Slovenia. LREC.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

# The University of Helsinki submissions to the WMT18 news task

**Alessandro Raganato, Yves Scherrer, Tommi Nieminen,**
**Arvi Hurskainen** and **Jörg Tiedemann**
Department of Digital Humanities
University of Helsinki

## Abstract

This paper describes the University of Helsinki's submissions to the WMT18 shared news translation task for English-Finnish and English-Estonian, in both directions. This year, our main submissions employ a novel neural architecture, the Transformer, using the open-source OpenNMT framework. Our experiments couple domain labeling and fine tuned multilingual models with shared vocabularies between the source and target language, using the provided parallel data of the shared task and additional back-translations. Finally, we compare, for the English-to-Finnish case, the effectiveness of different machine translation architectures, starting from a rule-based approach to our best neural model, analyzing the output and highlighting future research.

## 1 Introduction

The University of Helsinki participated in the WMT 2018 shared task on news translation with seven primary submissions. While the main focus of our work lay on the English-to-Finnish translation direction, we also participated in the Finnish-to-English, English-to-Estonian and Estonian-to-English translation directions.

In 2017, the University of Helsinki participated in WMT with an in-house implementation of an attentional encoder-decoder architecture based on the Theano framework, called HNMT (Östling et al., 2017). Since then, the development of Theano has stopped, and various open-source Neural Machine Translation (NMT) toolkits based on alternative frameworks have been made available (Klein et al., 2017; Junczys-Dowmunt et al., 2018, *inter alia*). In parallel, a novel neural network architecture for machine translation, called Transformer, has been introduced (Vaswani et al., 2017). The Transformer follows the encoder-decoder paradigm, but does not use any recurrent layers. Instead, its architecture relies primarily on attention mechanisms, stacking on each layer multiple attention components. Preliminary experiments with the Transformer architecture and its implementation in OpenNMT-py (Klein et al., 2017) showed consistent performance improvements compared to our 2017 architecture. Consequently, we used this setup for our main WMT 2018 submissions. For English–Finnish, our submissions also include a rule-based system, an SMT system, and a NMT system making use of a morphological analyzer and generator.

This year's WMT news translation task contains a multilingual sub-track, which includes all models that make use of third language data. We trained a multilingual model with data coming from three languages, English, Finnish and Estonian and then fine-tuned on a single language pair. We also generated synthetic English–Estonian data by pivoting through Finnish.

Additionally, following recent approaches (Johnson et al., 2016; Tars and Fishel, 2018) we added a domain label to each input sentence, according to the data source. For example, each sentence from the Europarl corpus was prepended with the ⟨*EUROPARL*⟩ label. The overall idea of domain labelling is that data coming from different sources are of different quality and represent different genres and writing styles. In this way, the translation model can be informed of the data source without increasing the number of parameters.

## 2 English→Finnish

### 2.1 NMT models

We trained our systems on almost all parallel data made available by WMT: Europarl (Koehn, 2005), ParaCrawl[1], Rapid, as well as the WMT 2015 test and development sets. We did not use WikiHeadlines. For development and tuning of the system

[1] https://paracrawl.eu/

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 488–495
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64052

parameters, we used the WMT 2016 and 2017 test sets.

A common strategy is to create synthetic training data by back-translation (Sennrich et al., 2016a). For our WMT 2017 submission, we already used SMT to create 5.5M sentences of back-translated data from the Finnish news2014 and news2016 corpora. This year, we created another 5.5M sentences of back-translation from the Finnish news2014-news2017 corpora using our previous NMT system (Östling et al., 2017). The final submissions make use of both resources.

We applied the standard preprocessing pipeline consisting of tokenization,[2] normalization,[3] truecasing and byte-pair encoding (Sennrich et al., 2016b). Following Vaswani et al. (2017), we have used a joint BPE vocabulary of 37 000 units. Regarding domain labeling, we marked the development and test data from WMT 2015, 2016 and 2017 as ⟨*NEWS*⟩. This label is also used for the test sets.

## 2.2 NMT with morphological analysis and generation

We also submitted a neural machine translation model that uses a morphological analyzer, called TwoStepTransformer. TwoStepTransformer is an English to Finnish transformer-type NMT model trained with the Marian NMT framework (Junczys-Dowmunt et al., 2018), using the default transformer settings (corresponding to Google's original Transformer setup). The model differs from standard NMT models in that the Finnish corpus is analyzed with a morphological analyzer (FINTWOL by Lingsoft Inc.) and segmented into a sequence of interleaved lemmas and morphological tags. The output of the model is converted into surface forms in a separate, deterministic post-processing step.

A similar two-step approach has been found to improve English to Czech NMT (Tamchyna et al., 2017), probably due to alleviating data sparsity caused by morphological complexity. As Finnish is also a morphologically complex language, adapting this approach to Finnish should result in a similar improvement. Finnish is an agglutinative language with a high degree of allomorphy for root, inflectional and derivational morphemes. For instance, the plural affix is expressed as *t*, *i* or *j* depending on the morphological context, and it is common for root lexemes have more than two allomorphs (e.g. the lexeme with the meaning 'hand' has the allomorphs *käsi*, *käde*, *käte* and *kät*). This allomorphy greatly increases data sparsity if segmentation methods based on surface form splitting are used.

The annotation format used differs from the one in Tamchyna et al. (2017) in several aspects, the most important of which is that the morphological tags are not complex, multicategory tags that are interleaved one-to-one with lemmas. Instead, each lemma token can be followed by zero or more morphological tags, each corresponding to a non-default value in a single morphological category:

```
komissio
tiedottaa FINTWOL_PAST
neuvosto FINTWOL_ALL EU FINTWOL_GEN
ja
Marokko FINTWOL_GEN
kalastus LS_PRECOMPOUND
kumppanuus LS_PRECOMPOUND
sopimus FINTWOL_PTV
koskeva FINTWOL_ELA FINTWOL_PL
kahden LS_PRECOMPOUND
välinen FINTWOL_ELA FINTWOL_PL
neuvottelu FINTWOL_ELA FINTWOL_PL
.
```

The first lemma *komissio* is the only one without any morphological tags, the rest of the lemmas are trailed by one or more tags. Tags are only provided if the value of a morphological category differs from the default value, so this means that the lemma *komissio* has the default value for number (singular) and case (nominative). The lemma *tiedottaa* is a verb lemma (lemma form indicates word class so no explicit word class annotation is required), and it has the tag FINTWOL_PAST, indicating that it has the non-default value PAST for the tense category (default is present tense). Several noun lemmas have non-default case and number values, for example *neuvottelu*, which has allative case and plural number. The LS_PRECOMPOUND tag indicates the lemma is part of a compound word.

There are several reasons for using implicit default morphological categories:

1. Explicitly defining each tag would lead to very long target sentences.

2. Having separate tags for each category theoretically allows for more generalization than complex multi-category tags. For instance, case generalizations could be learned from both singular and plural contexts.

3. Languages generally have morphological categories where the most common value has no

[2] We modified the Moses tokenizer to prevent it from splitting word-internal colons that occur regularly in Finnish.

[3] Normalization is applied to English only and consists of resolving common contractions such as *isn't*, *we'll* etc.

explicit morpheme, so segmenting with implicit common values makes the segmented text structurally more similar to natural language.

The morphological segmentation (which includes compound splitting) decreases the amount of token types in the corpus significantly (from over a million to about 300,000 for the bilingual WMT data), but there are still too many token types for efficient NMT training, due to foreign language words, incorrectly spelled words, numbers, codes, character corruption and other out-of-vocabulary tokens. To lower the type count to a manageable level, the annotated corpus is further segmented using BPE. As the model outputs a BPE sequence of lemmas and morphological tags, producing the final translation is more complex than simple concatenation of subword units. First the BPE tags are joined and then the surface forms are generated using the FINTWOL generation functionality, which takes as input lemmas and morphological tags and output all compatible surface forms. The default tags are automatically added for lemmas which do not have explicit tags. Heuristics are used to select a surface form if several possibilities are generated.

The submitted model was trained on the bilingual and back-translated data, as adding the back-translated data greatly improved the quality of the translations.

### 2.3 Rule-based MT

Hurskainen and Tiedemann (2017) propose a rule-based machine translation system for English–Finnish. During the past year, the rule-based MT system has been developed in several ways. In addition to the usual debugging and rule testing, also some major structural changes have been made. Below we will discuss the latter type of problems.

**Translating locative expressions:** While English uses prepositions for marking location, Finnish uses locative cases. English has a bewildering number of prepositions for this purpose. At least the following preposition are used: *in, on, at, with, by, to, into, for, of, from, over, through*, and *around*. Finnish uses one of the six locative cases for translating such structures.

Locative cases can be classified into two groups, which are termed as internal (inessive, elative, and illative) and external (adessive, ablative, and allative) locatives. Associating the English locative preposition with one of the Finnish locative cases would require several rules with a varying number of constraints. In the current implementation, the Finnish locative cases are handled in two phases. In the first phase, we only consider what type (no movement, movement from, movement to) the location is, without considering whether it is internal or external.

```
"<he>" "he" { hän } %SUBJ HUM OUT PRON PERS SG3 NOM
"<sent>" "send" { lähetti } %+FMAINV O-ACC O-LOC3 V
    PAST SG
"<letter>" "letter" { kirjeen } %OBJ DEF N SG ACC
"<to>" "to" { M-LOC3 } %ADVL PREP
"<hospital>" "hospital" { sairaalaan } %<P ACE IN
    DEF N SG ILL
"<.>" "." { . }
```

```
"<he>" "he" { hän } %SUBJ HUM OUT PRON PERS SG3 NOM
"<sent>" "send" { lähetti } %+FMAINV O-ACC O-LOC3 V
    PAST SG
"<letter>" "letter" { kirjeen } %OBJ DEF N SG ACC
"<to>" "to" { M-LOC3 } %ADVL PREP
"<me>" "i" { minulle } %<P HUM OUT PRON PERS SG ALL
```

We see that in both sentences the preposition *to* has the tag M-LOC3. This stands for illative and allative. The head of the preposition decides which of the cases is selected. If the noun has the tag OUT, then allative is selected. If it has the tag IN or no locative tag, then inessive is selected. The same process applies to the two other locative case pairs (inessive/adessive and elative/ablative).

A special case of using locatives are the Finnish place names. No formal rules can be constructed for producing correct locative inflection. Therefore, we have to tag each place name separately. We use internal inflection as default and provide names using external inflection with the tag OUT.

**Translating proper names and acronyms:** There are two major problems in dealing with proper names and acronyms. One concerns the question whether the proper name or acronym should be translated or not. The other problem concerns the handling of uppercase letters. The proper names with translation should be listed in the lexicon or handled as an MWE. It is assumed that a non-sentence-inital word with capital initial is a proper name, and possibility to such an interpretation is provided by adding a separate entry with a tag PROP-CAND. If it is not listed in the lexicon, it is interpreted as a proper name. Such words which have also another interpretation in the language are problematic. Many person names belong to this category. Attested cases with both interpretations (i.e. normal translation and proper name) are listed in the rule system. Then, using context sensitive rules, the PROP-CAND interpretation is selected or removed.

**Translating subject and object:** The default case of the subject in Finnish is nominative, but also other cases, such as adessive, genitive, elative, ablative, and illative, occur. Rules are needed only for the special cases. This is implemented by providing the respective verb with a tag showing the case of the subject. Otherwise the subject case is always nominative. The direct object has three cases, partitive, genitive accusative and nominative accusative. The last one is used in special cases such as the object of imperative verb form and some modal verbs. Partitive and genitive accusative dominate as object case. Part of verbs require always partitive, and some others require the genitive accusative case. However, most verbs are such that they may have either of the object cases. They are not alternatives, however, because the context defines the case in each situation. There is the general trend that if the object is indefinite plural, it is in partitive.

More details of the system are described in Hurskainen (2018a,b,c,d).

### 2.4 SMT

As a contrastive system, we also reactivated our Statistical Machine Translation (SMT) system submitted at WMT 2016 (Tiedemann et al., 2016). The system was not retrained and it may thus suffer from poor lexical coverage on recent test data. Our main motivation for including this baseline was to have an SMT submission for the Finnish morphology test suite.

## 3 Finnish→English

We only submitted a standard NMT transformer model with domain labeling for this translation direction. Parallel data and preprocessing steps are identical as for English-to-Finnish. For back-translation, we use 2M sentences from the English news2015 produced with an SMT system, plus another 6.7M sentences from English news2007–news2017 produced with HNMT (Östling et al., 2017).

During the test phase, we discovered that several source lines, in particular in the Finnish test data, consisted of more than one sentence. As our translation systems were trained mostly on single sentences, they tended to stop the translation process after translating the first sentence of the line, leaving the remaining sentences untranslated. In order to avoid this, we applied a simple sentence splitting script to the test set and translated the split sentences separately. According to the output of the sentence splitter, 298 sentences of the Finnish source and 13 sentences of the English source were affected. We applied sentence splitting to both files; while this increased BLEU scores by 0.5 points on Finnish-to-English, it did not affect the BLEU scores of English-to-Finnish translation.

## 4 English–Estonian

We also participated in the English–Estonian task, in both directions. We used all available parallel data for training: Europarl, ParaCrawl, and Rapid. We used the 2018 dev set for system development and parameter tuning. We applied the same preprocessing steps as for English–Finnish, using again a shared vocabulary of 37 000 BPE units. Regarding domain labeling, no parallel data with the ⟨*NEWS*⟩ label was provided in this setup. Therefore, we labelled the test source data with ⟨*EUROPARL*⟩, which we found to be the most reliable of the three data sources. For comparison, we also tested a model without domain labels (comparative results are given in Section 6).

For our English-to-Estonian submission, we created back-translations using a simpler translation model. This model was based on the Transformer architecture and was trained on a subset of parallel data filtered through a language identification tool, with 20 000 BPE units. We used this model to translate parts of the monolingual BigEst corpus to English; 6.3M back-translations sentences were obtained.

For the Estonian-to-English submission, we also generated back-translations using a simple translation model, as described above, translating parts of the monolingual English news2007–news2017 corpora; 5.2M back-translation sentences were produced in this way.

## 5 Multilingual models

As Estonian is closely related to Finnish, we experimented with multilingual models containing both languages as well as English. For this experiment, we included all available parallel data in all directions. Following Johnson et al. (2016), we used language labels to indicate the target language coupled with the domain labels, as introduced above. The only other change in preprocessing is the use of 50 000 (instead of 37 000) joint BPE units, as they now need to cover three languages instead of

|  | Parallel | +Back | +Back +Synth |
|---|---|---|---|
| **Et → En** | 2,178,025 | 7,356,697 | 8,942,157 |
| **En → Et** | 2,178,025 | 8,435,413 | 10,020,873 |
| **Fi → En** | 3,136,265 | 11,918,402 | – |
| **En → Fi** | 3,136,265 | 14,198,188 | – |

Table 1: Number of training sentences, with and without back-translation (Back) and synthetic data (Synth).

two. In this way, even though Estonian has no parallel news data, the model will see the news label in the Finnish data. Inspired by Zoph et al. (2016), we first train the multilingual model with all languages in all directions, and then fine-tune it on each specific language pair.

### 5.1 Synthetic data

Another way to take advantage of the close etymological relationship between Estonian and Finnish is to create synthetic training data (Tiedemann, 2012). We explored this option in the following setup:

1. Train a character-level seq2seq system for Finnish-to-Estonian, using the Europarl and EUbookshop (Skadiņš et al., 2014) corpora.
2. Translate the Finnish side of the parallel English–Finnish corpus to Estonian.
3. Combine the Estonian and English parts of the corpus and use this dataset as back-translations to train the final system.

We were able to process 1.5M sentences using this approach. These sentences complemented the other training data, consisting of parallel data and direct English–Estonian back-translations.

## 6 Experiments

In this section we detail the setup of our experiments. We first describe the size of the training data and the details of the training; we then report and discuss the performance of each model according to the BLEU score as reported on the online evaluation matrix[4].

Table 1 shows the statistics on the number of training sentences. The backtranslations allow us to more than triple the original size of the training data for all the directions. We trained our models for 20 epochs, evaluating each of them on the development set after every epoch, taking the best iteration as final model. As hyper-parameters, we used the *base* version of the Transformer architecture, following the suggestion of the OpenNMT-py tool,[5] including a shared word embedding space between encoder and decoder among others. Unlike last year, we did not include any averaging or ensembling techniques.

|  |  |  | Et→En | En→Et |
|---|---|---|---|---|
| **HY-NMT Baseline** |  |  | 21.6 | 16.7 |
| **+Label** |  |  | 20.3 | 17.6 |
|  | **+Back** |  | **26.5** | – |
| **+Label** | **+Back** |  | 25.4 | **21.8*** |
|  | **+Back** | **+Synth** | **26.5*** | – |
| **+Label** | **+Back** | **+Synth** | 25.0 | 21.0 |
| **HY-NMT Multilingual** |  |  | – | – |
| **+Label** |  |  | 26.4 | 20.8 |

Table 2: BLEU-cased scores on newstest2018 for the English–Estonian language pair in various configurations using domain labels (Label), backtranslated data (Back), or synthetic data (Synth). Our primary submissions are marked with *.

**English–Estonian results.** Table 2 shows the performance of our models for the English–Estonian language pair.[6] In general, the best models include back-translation and synthetic data, improving the BLEU score by around 4 points. The domain labels help when translating into Estonian, while they slightly hurt the performance when translating into English. This behavior could be explained by the different nature of the two languages, Estonian being a morphologically rich language, it could benefit from having a source label indicating good quality translations even if they come from a different domain. As concerns our multilingual model, it achieves results close to our best score, specially for the Estonian-to-English direction. We recall that this model also uses domain labels, and this suggests that, in this case, the Finnish–English data are indeed helpful to achieve a better BLEU score for the Estonian-to-English language pair.

**English–Finnish results.** Table 3 shows the performance of our models for the English–Finnish language pair. Here, all of our basic Transformer

[4] http://matrix.statmt.org/

[5] http://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model

[6] HY stands for *Helsingin Yliopisto*, i.e. University of Helsinki, not for *hybrid*.

| | Fi→En | En→Fi |
|---|---|---|
| **Transformer +Label** | 19.8 | 15.3 |
| **Transformer +Back +Label** | **23.3*** | **17.8*** |
| **Multilingual +Back +Label** | 20.6 | 14.9 |
| **TwoStep +Back** | – | 14.5* |
| **Seq2Seq +Back +Label** | – | 12.1 |
| **HY-SMT +Back** | – | 10.5* |
| **HY-AH (rule based)** | – | 6.4* |

Table 3: BLEU-cased scores on newstest2018 for the English–Finnish language pair for various system architectures. Our primary submissions are marked with *.

models included domain labeling, motivated by the fact that news data are present in the training data, and also by the consistent performance improvements observed in initial experiments. Overall, back-translations again improve the BLEU scores for both directions. The multilingual model achieves lower scores than the standard bilingual model, suggesting that the Estonian data do not provide useful complementary information, in particular because the Estonian data set is rather small compared to its Finnish counterpart and comes from exactly the same source. Finally, we also compare our transformer-based models to other machine translation paradigms. Table 3 reports the BLEU scores of the rule based system described in Section 2.3, the SMT system (Section 2.4) and an additional 2-layer sequence-to-sequence model (Bahdanau et al., 2014) trained on the same data as the Transformer models. Clearly, the Transformer paradigm achieves the best BLEU scores. Overall, our best English-to-Finnish model reaches the second position in the online ranking using automatic evaluation metrics. Finally, in the manual evaluation of the official results of the WMT18 News Translation task (Bojar et al., 2018), our best system shared first place in both English-to-Finnish and Finnish-to-English translation directions.

### 6.1 English-to-Finnish analysis

To complement the results, we additionally carried out an analysis of the output of our best English-to-Finnish system.

**Document knowledge.** One of the common mistakes is related to pronouns, especially when their antecedents are located in other sentences. As our systems are trained on isolated sentences, it is hard to predict the right pronoun when it refers to a previous sentence. Moreover, more context would help to better understand the semantics of the sentence. For example, considering the following translation:

(1) EN: “After burying the bodies, the military came looking for me,” he says.
FI: “Sotilaat etsivät minut käsiinsä uhrien hautaamisen jälkeen”, hän sanoo.

the word *bodies* has been translated as *victims*, which only makes sense if you know the document context where bodies were those of demonstrators.

**World knowledge.** We found out that some test set translations contain information based on world knowledge” outside of the actual text, and so the system being trained without any external knowledge fails to output the most appropriate translation. For example, in the sentences:

(2) EN: “Americans appreciate this as well as anyone - hence the carefully stage-manged toppling of Saddam Hussein in Firdos square in Baghdad in 2003.”
FI: “Amerikkalaiset tietävät sen yhtä hyvin kuin muutkin: irakilaiset kaatoivat yhdessä amerikkalaisten sotilaiden kanssa Saddam Husseinin patsaan Firdosin aukiolla vuonna 2003.”

the literal translation of the Finnish sentence would be: *“The Americans know it as well as others: the Iraqi toppled together with American soldiers Saddam Hussein’s statue in Firdos square in 2003.”*, leaving out Baghdad and introducing Iraqi in this case.

Finally, a number of errors were related to the different structure and ordering of the words of the two languages. It seems like the 2018 test set is translated more freely and document-oriented than in previous years, which explains the overall low BLEU scores compared to the last year’s competition.

## 7 Conclusions

In this paper, we reported the University of Helsinki submissions for the WMT18 news translation task. We participated in the English–Finnish and English–Estonian language pairs, training the novel neural architecture, the Transformer, with the OpenNMT tool, using BPE segmentation, a joint source-target vocabulary and domain labeling. Additionally, we introduced a multilingual model trained on all our data sets, fine-tuning it on each language pair. Our best systems are trained on the provided parallel data augmented with large amounts of back-translations, achieving top rank results for the English–Finnish language pair. We also carried out further analyses on the English-to-Finnish direction, showing the performance of different machine translation paradigms and highlighting common mistakes that prevented a higher translation quality.

## Acknowledgments

The work in this paper is supported by the Academy of Finland through project 314062 from the ICT 2023 call on Computation, Machine Learning and Artificial Intelligence. We would also like to acknowledge NVIDIA and their GPU grant.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Arvi Hurskainen. 2018a. Comparative and superlative in English to Finnish machine translation. Technical Reports in Language Technology 26, University of Helsinki, http://www.njas.helsinki.fi/salama.

Arvi Hurskainen. 2018b. Implementing location in English to Finnish machine translation. Technical Reports in Language Technology 27, University of Helsinki, http://www.njas.helsinki.fi/salama.

Arvi Hurskainen. 2018c. Proper names and acronyms in English to Finnish machine translation. Technical Reports in Language Technology 28, University of Helsinki, http://www.njas.helsinki.fi/salama.

Arvi Hurskainen. 2018d. Subject and object case in English to Finnish machine translation. Technical Reports in Language Technology 29, University of Helsinki, http://www.njas.helsinki.fi/salama.

Arvi Hurskainen and Jörg Tiedemann. 2017. Rule-based machine translation from English to Finnish. In *Proceedings of the Second Conference on Machine Translation*, pages 323–329, Copenhagen, Denmark. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Robert Östling, Yves Scherrer, Jörg Tiedemann, Gongbo Tang, and Tommi Nieminen. 2017. The Helsinki neural machine translation system. In *Proceedings of the Second Conference on Machine Translation*, pages 338–347, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne. 2014. Billions of parallel words for free: Building and using the EU Bookshop corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ales Tamchyna, Marion Weller-Di Marco, and Alexander M. Fraser. 2017. Modeling target-side inflection in neural machine translation. *CoRR*, abs/1707.06012.

Sander Tars and Mark Fishel. 2018. Multi-domain neural machine translation. *arXiv preprint arXiv:1805.02282*.

Jörg Tiedemann. 2012. Character-based pivot translation for under-resourced languages and domains. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 141–151. Association for Computational Linguistics.

Jörg Tiedemann, Fabienne Cap, Jenna Kanerva, Filip Ginter, Sara Stymne, Robert Östling, and Marion Weller-Di Marco. 2016. Phrase-based SMT for Finnish with more data, better models and alternative alignment and translation tools. In *Proceedings of the First Conference on Machine Translation*, pages 391–398, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575.

# The RWTH Aachen University Supervised Machine Translation Systems for WMT 2018

**Julian Schamper, Jan Rosendahl, Parnia Bahar,**
**Yunsu Kim, Arne Nix and Hermann Ney**
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
`<surname>@i6.informatik.rwth-aachen.de`

## Abstract

This paper describes the statistical machine translation systems developed at RWTH Aachen University for the German→English, English→Turkish and Chinese→English translation tasks of the *EMNLP 2018 Third Conference on Machine Translation* (WMT 2018). We use ensembles of neural machine translation systems based on the Transformer architecture. Our main focus is on the German→English task where we scored first with respect to all automatic metrics provided by the organizers. We identify data selection, fine-tuning, batch size and model dimension as important hyperparameters. In total we improve by 6.8% BLEU over our last year's submission and by 4.8% BLEU over the winning system of the 2017 German→English task. In English→Turkish task, we show 3.6% BLEU improvement over the last year's winning system. We further report results on the Chinese→English task where we improve 2.2% BLEU on average over our baseline systems but stay behind the 2018 winning systems.

## 1 Introduction

In this paper we describe the supervised statistical machine translation (SMT) systems developed by RWTH Aachen University for the news translation task of the *EMNLP 2018 Third Conference on Machine Translation* (WMT 2018). We use ensembles of neural machine translation systems to participate in the German→English, English→Turkish and Chinese→English tasks of the WMT 2018 evaluation campaign.

For this year's WMT we switch towards the Transformer architecture (Vaswani et al., 2017) implemented in Sockeye (Hieber et al., 2017). We experiment with different selections from the training data and various model configurations.

This paper is organized as follows: In Section 2 we describe our data preprocessing. Our translation software and baseline setups are explained in Section 3. The results of the experiments for the various language pairs are summarized in Section 4.

## 2 Preprocessing

For all our experiments on German, English and Turkish we utilize a simple preprocessing pipeline which consists of minor text normalization steps (e.g. removal of some special UTF-8 characters) followed by tokenization from Moses (Koehn et al., 2007) and frequent casing from the Jane toolkit (Vilar et al., 2010). The Chinese side is segmented using the `Jieba4` segmenter [1] except for the `Books 1-10` and `data2011` data sets which were already segmented as mentioned in (Sennrich et al., 2017).

We apply byte-pair encoding (BPE) to segment words into subword units for all language pairs (Sennrich et al., 2016b). Our BPE models are trained jointly for the source and the target language with the exception of the Chinese→English task. For every language pair we use the parallel data to train the BPE operations, excluding any synthetic data and the ParaCrawl corpus of the German→English task. To reduce the number of rare events we apply a vocabulary threshold of 50 as described in (Sennrich et al., 2017) in all our German→English systems. We end up with vocabulary sizes of 45k and 34k for German and English respectively if 50k joint merge operations are used.

## 3 MT Systems

All systems submitted by RWTH Aachen are based on the Transformer architecture imple-

[1] `https://github.com/fxsjy/jieba`

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 496–503
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64053

mented in the Sockeye sequence-to-sequence framework for Neural Machine Translation. Sockeye is built on the Python API of MXNet (Chen et al., 2015).

In the Transformer architecture both encoder and decoder consist of stacked layers. A layer in the encoder consists of two sub-layers: a multi-head self-attention layer followed by a feed forward layer. The decoder contains an additional multi-head attention layer that connects encoder and decoder. Before and after each of these sub-layers preprocessing respectively postprocessing operations are applied. In our setup layer normalization (Ba et al., 2016) is applied as preprocessing operation while the postprocessing operation is chosen to be dropout (Srivastava et al., 2014) followed by a residual connection (He et al., 2016).[2] For our experiments we use 6 layers in both encoder and decoder and vary the size of their internal dimension. We set the number of heads in the multi-head attention to 8 and apply label smoothing (Pereyra et al., 2017) of 0.1 throughout training.

We train our models using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001 (for En→Tr and Zh→En) respectively 0.0003 (for De→En). A warmup period with constant or increasing learning rate was not used. We employ an annealing scheme that scales down the learning rate if no improvement in perplexity on the development set is seen for several consecutive evaluation checkpoints. During training we apply dropout ranging from 0.1 to 0.3. All batch sizes are specified on the token level and are chosen to be as big as the memory of the GPUs allows. In case of the utilization of multiple GPUs we use synchronized training, i.e. we increase the effective batch size, which seems to have better convergence properties (Popel and Bojar, 2018).

The weights of embedding matrices and the projection layer prior to the softmax layer are not shared in our architecture and for all translation runs a beam size of 12 is used.

## 4 Experimental Evaluation

In this section we present our results on the three tasks we participated in, with the primary focus on building a strong system for the German→English system.

| Data Resource | # lines |
|---|---|
| WMT 2018 (standard parallel data) | 5.9M |
| ParaCrawl (filtered 50%) | 18.2M |
| NewsCrawl 2015 (UEDIN WMT16) | 4.2M |
| NewsCrawl 2017 (random 50%) | 13.4M |
| newstest2008–2014 (for fine-tuning) | 19.1k |

Table 1: Number of parallel sentence pairs or monolingual sentences of our different training data resources (De→En). Our strongest systems from Table 3 use all these resources.

For evaluation we use `mteval-v13a` from the Moses toolkit (Koehn et al., 2007) and `TERCom`[3] to score our systems on the BLEU (Papineni et al., 2002) respectively TER (Snover et al., 2006) measures. In addition we report CTER scores[4] (Wang et al.). All reported scores are given in percentage and the specific options of the tools are set to be consistent with the calculations of the organizers.

### 4.1 German→English

In most experiments for the German→English task we use a subset of the data resources listed in Table 1. All models use the Transformer architecture as described in Section 3. Our baseline model is very similar to the ”base” Transformer of the original paper (Vaswani et al., 2017), e.g. $d_{\text{model}} = 512$ and $d_{\text{ff}} = 2048$, however we do not use weight-tying.

Throughout our experiments we analyze various aspects of our experimental setup (e.g. several data conditions or the model size). We evaluate our models every 20k iterations and select the best checkpoint based on BLEU calculated on our development set `newstest2015` afterwards. To handle all different variations in a well organized way, we use the workflow manager Sisyphus (Peter et al., 2018).

In Table 2 we carefully analyze different data conditions. We can see that the Transformer model with 20k BPE merging operation already beats our last year’s final submission by 1.4% BLEU. The Transformer model was trained using the standard parallel WMT 2018 data sets (namely Europarl, CommonCrawl, NewsCommentary and Rapid, in total 5.9M sentence pairs) as well as the 4.2M sen-

[2] Note that this is by now also the default behavior of the Tensor2Tensor implementation https://github.com/tensorflow/tensor2tensor, differing from the original paper.

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

[4] https://github.com/rwth-i6/CharacTER

| | Systems | BPE | newstest2015 (dev) | | |
|---|---|---|---|---|---|
| | | | BLEU | TER | CTER |
| 1 | RWTH WMT 2017 | | 32.0 | - | - |
| 2 | Transformer | 20k | 33.4 | 52.7 | 48.4 |
| 3 | + ParaCrawl | 20k | 30.3 | 57.8 | 51.9 |
| 4 | Transformer | 50k | 33.9 | 52.5 | 47.9 |
| 5 | + ParaCrawl | 50k | 31.3 | 56.5 | 50.7 |
| 6 | + Oversample | 50k | 31.6 | 56.3 | 50.5 |
| 7 | + Filtering | 50k | 34.7 | 51.7 | 46.9 |

Table 2: Baseline results and analysis of data conditions (De→En). Our baseline starts of with the standard WMT 2018 training data excluding ParaCrawl but including already backtranslated NewsCrawl 2015. ”Filtering” refers to filtering ParaCrawl only (50% LM driven).

tence pairs of synthetic data created in (Sennrich et al., 2016a). Last year's submission is an ensemble of several carefully crafted models using an RNN-encoder and decoder which was trained on the same data plus 6.9M additional synthetic sentences (Peter et al., 2017).

We try 20k and 50k merging operations for BPE and find that 50k performs better by 0.5% to 1.0% BLEU. Hence, we use this for all further experiments. As Table 2 shows, just adding the new ParaCrawl corpus to the existing data hurts the performance by up to 3.1% BLEU.

To counter this effect we oversample CommonCrawl, Europarl and NewsCommentary with a factor of two. Rapid and the synthetic news data are not oversampled. As we can observe in Row 6 of Table 2 this gives a minor improvement, but is not enough to counter the negative effects from adding ParaCrawl. Therefore we train a 3-gram language model on the monolingual English NewsCrawl2007-2017 data sets using KenLM (Heafield, 2011) to rank the corpus and select the best 50% of sentence pairs. Together with oversampling this yields an improvement of 3.4% BLEU over the naive concatenation of all training data and 0.8% BLEU over the corresponding system that does not use ParaCrawl at all.

Using the best data configuration described we start to use multiple GPUs instead of one and increase the model size. Each GPU handles a share of the data and the update steps are synchronized, such that the effective batch size is increased. As before we choose the batch size on word level in such a way that the memory of all GPUs is fully used. Note that due to time constraints and the size of the models the reported results are taken from models which did not yet fully converge. Each model in Table 3 is trained using 4 GPUs for close to 8 days.

First we double the dimension of the model to $d_{\text{model}} = 1024$. As can be seen from Table 3, together with the increased batch size, this yields a major improvement of 1.2% BLEU on newstest2015.

Using a basic English→German system we backtranslate 26.9M sentences from the NewsCrawl 2017 monolingual corpus. This system uses the same transformer configuration as used for the baseline De→En system and is trained on the standard parallel WMT 2018 dataset (5.9M sentence pairs). It achieves 28.7% BLEU and 38.9% BLEU on newstest2015 and newstest2018 respectively. After experimenting with several thresholds we added half of the backtranslated data (randomly selected) to our training data which gave us 0.5% BLEU extra on the development set. Even though the system is trained on 17.6M synthetic news sentences from NewsCrawl 2015 (4.2M) and NewsCrawl 2017 (13.4M), fine-tuning on old test sets (newstest2008 to newstest2014) improves it by 0.6% BLEU on newstest2015 and 1.0% BLEU on newstest2017. We set the checkpoint frequency down to 50 updates only and select the best out of 14 fine-tuned checkpoints (selected on newstest2015). Overall we find it beneficial to match the learning conditions which are present for the checkpoint which is fine-tuned: Especially important seems to be the usage of a similar learning rate in contrast to using the comparably high initial learning rate (0.0003).

Adding an extra layer to encoder and decoder did not change the performance of the system significantly. However the model was helpful in the final ensemble. Similarly increasing the dimension of the hidden size of the feed forward layers to 4096 and setting the number of attention heads to 16 barely changed the performance of the system. It turns out to be helpful if we double the batch size of the model. Because the GPUs available to us can not handle bigger batches we increased the effective batch size further by accumulating gradient updates before applying them. The resulting system shown in Table 3 Row 7 is the best single system provided by RWTH Aachen for

| | | newstest2015 (dev) | | | newstest2017 | | | newstest2018 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Systems | BLEU | TER | CTER | BLEU | TER | CTER | BLEU | TER | CTER |
| 1 | Baseline (Table 2 Row 7) | 34.7 | 51.7 | 46.9 | 36.4 | 50.7 | 47.1 | 44.5 | 41.4 | 39.6 |
| 2 | + $d_{\text{model}}$ =1k + 4GPUs | 35.9 | 50.9 | 46.2 | 37.6 | 49.6 | 45.9 | 46.1 | 40.4 | 38.5 |
| 3 | + NewsCrawl17 (50%) | 36.4 | 50.1 | 45.7 | 38.3 | 49.2 | 46.3 | 46.8 | 39.6 | 38.2 |
| 4 | + fine-tuning | 37.0 | 49.5 | 45.5 | 39.3 | 48.1 | 45.2 | 47.5 | 38.9 | 37.7 |
| 5 | + 7th layer | 37.2 | 49.6 | 45.3 | 39.3 | 48.0 | 45.0 | 47.5 | 39.0 | 37.8 |
| 6 | + $d_{\text{ff}}$ =4k + 16 heads | 37.0 | 49.8 | 45.6 | 39.0 | 48.4 | 45.1 | 47.5 | 38.8 | 37.6 |
| 7 | + GradAcc=2 | 37.1 | 49.2 | 45.3 | 39.5 | 48.0 | 45.0 | 47.6 | 38.8 | 37.6 |
| 8 | Ensemble [4,5,7] | 37.5 | 49.1 | 44.9 | 39.9 | 47.6 | 44.6 | 48.4 | 38.1 | 36.9 |

Table 3: Main results for the German→English task (Rows 7 and 8 show the submitted system). Note that using multiple GPUs does not only result in a higher data throughput but also multiplies the effective batch size and therefore affects the convergence. However if only one GPU is available the results could be still reproduced by using just gradient accumulation.

| | nt2015 | nt2017 | | |
|---|---|---|---|---|
| Systems | BLEU | BLEU | TER | CTER |
| Winner 2017 | - | 35.1 | 52.4 | 48.9 |
| RWTH 2017 | 32.0 | 33.1 | 54.2 | - |
| RWTH 2018 | 37.5 | 39.9 | 47.6 | 44.6 |

Table 4: Comparison with last years' submissions on `newstest2015+2017` (De→En). The winning system of 2017 was submitted by UEDIN. Missing scores are due to inconsistent calculations or unavailability.

the German→English task.

Because checkpoint averaging helped in the past we tried several versions based on last or best checkpoints of different distances but no version turned out to be helpful in our case.

Finally model ensembling brought performance up to 37.5% BLEU and 39.9% BLEU on `newstest2015` and `newstest2017`. Overall we achieved an improvement of 2.8% and 3.5% BLEU over our baseline.

Table 4 shows that we improved our system by 6.2% BLEU on average on `newstest2015+2017` since previous year and by 4.8% BLEU on `newstest2017` over the winning system of 2017 (Sennrich et al., 2017).

### 4.2 English→Turkish

The English→Turkish task is in a low-resource setting where the given parallel data consists of only around 200k sentences. We therefore apply dropout to various parts of our Transformer model: attention/activation of each layer, pre/post-processing between the layers, and also embedding—with a dropout probability of 0.3. This gives a strong regularization and yields 2.6% BLEU improvement compared to the baseline in `newstest2018` (Row 2 of Table 5).

Although the English and Turkish languages are from different linguistic roots, we find that the performance is better by 4.5% BLEU in `newstest2018` when sharing their vocabularies by tying the embedding matrices (Row 3 of Table 5). They are also tied with the transpose of the output layer projection as done in (Vaswani et al., 2017). We accordingly use BPE tokens jointly learned for both languages (20k merge operations). Since the training signals are weak from the given data, we argue that this kind of parameter sharing helps to avoid overfitting and copy proper nouns correctly.

Checkpoint frequency is set to 4k. Other model parameters and training hyperparameters are the same as described in Section 3.

Table 5 also shows results with back-translated data from Turkish News Crawl 2017 (Row 4, +3.8% BLEU in `newstest2018`). Using more than 1M sentences of back-translations does not help, which might be due to the low quality of back-translations generated with a weak model (trained only with 200k parallel sentences). Note that we oversample the given parallel data to make the ratio of the parallel/synthetic data 1:1. An ensemble of this setup with four different random seeds shows a slight improvement up to 0.2% BLEU (Row 4 vs. 6).

Finally, we fine-tune the models with `newstest2016+2017` sets to adapt to the news domain. We set the learning rate ten times lower (0.00001) and the checkpoint frequency to 100. Dropout rate is reduced to 0.1 for a fast adaptation. This provides an additional boost of

| | Systems | newsdev2016 (dev) BLEU | TER | CTER | newstest2017 BLEU | TER | CTER | newstest2018 BLEU | TER | CTER |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Baseline | 6.8 | 93.0 | 71.4 | 7.4 | 91.6 | 76.7 | 7.1 | 92.1 | 73.0 |
| 2 | + dropout 0.3 | 9.3 | 84.9 | 67.0 | 10.3 | 83.4 | 73.4 | 9.7 | 84.5 | 68.0 |
| 3 | + weight tying | 14.2 | 76.9 | 59.7 | 15.8 | 74.5 | 61.6 | 14.2 | 77.3 | 62.1 |
| 4 | + BT 1M sents | 16.7 | 72.3 | 56.3 | 20.0 | 68.0 | 56.8 | 17.0 | 72.3 | 58.5 |
| 5 | + fine-tuning | 17.6 | 71.2 | 56.6 | 25.0 | 62.1 | 53.7 | 17.7 | 71.3 | 58.5 |
| 6 | Ensemble 4x [4] | 16.7 | 71.8 | 56.1 | 20.1 | 67.7 | 56.7 | 17.2 | 72.0 | 58.3 |
| 7 | Ensemble 4x [5] | 17.7 | 70.6 | 55.8 | 25.1 | 62.1 | 53.1 | 18.0 | 71.0 | 58.0 |

Table 5: English→Turkish results. Row 6 is the submitted system.

0.7% BLEU for the single model (Row 4 vs. 5) and 0.8% BLEU for the ensemble (Row 6 vs. 7) in newstest2018.

### 4.3 Chinese→English

We use all available parallel data totaling 24.7M sentence pairs with 620M English and 547M Chinese words and follow the preprocessing described in Section 2. We then learn BPE with 50k merge operations on each side separately. newsdev2017 and newstest2017 containing 2002 and 2001 sentences are used as our development and test sets respectively. We also report results on newstest2018 with 3981 samples. We remove sentences longer than 80 subwords. We save and evaluate the checkpoints according to the BLEU score on the development set every 10k iterations.

In order to augment our training data, we back-translate the NewsCrawl2017 monolingual corpus consisting of approximately 25M samples using a En→Zh NMT system resulting in a total of 49.5 sentence pairs for training. The En→Zh NMT model is based on the RNN with attention encoder-decoder architecture (Bahdanau et al., 2014) implemented in Returnn[5] (Zeyer et al., 2018). The network is similar to (Bahar et al., 2017) with 4-layer of bidirectional encoders using long-short term memory cells (LSTM) (Hochreiter and Schmidhuber, 1997). We apply a layer-wise pre-training scheme that leads to both better convergence and faster training speed during the initial pre-train epochs (Zeyer et al., 2018). We start using only the first layer in the encoder of the model and add new layers during the training progress. We apply a learning rate scheduling scheme, where we lower the learning rate if the perplexity on the development set does not improve anymore.

For Zh→En, we run different Transformer configurations which differ slightly from the model described in Section 3. Our aim is to investigate the effect of various hyperparameters especially the model size, the number of layers and the number of heads. According to the total number of parameters, we call these models as below:

- **Transformer base**: a 6-layer multi-head attention (8 heads) consisting of 512 nodes followed by a feed forward layer equipped with 1024 nodes both in the encoder and the decoder. The total number of parameters is 121M. Training is done using mini-batches of 3000.

- **Transformer medium**: a 4-layer multi-head attention (8 heads) consisting of 1024 nodes followed by a feed forward layer equipped with 4096 nodes both in the encoder and the decoder. The total number of parameters is 271M. Training is done using mini-batches of 2000.

- **Transformer large**: a 6-layer multi-head attention (16 heads) consisting of 1024 nodes followed by a feed forward layer equipped with 4096 nodes both in the encoder and the decoder. The total number of parameters is 330M. Training is done using mini-batches of 6500 on 4 GPUs.

The results are shown in Table 6. Note that all models are trained using bilingual plus synthetic data. Comparing the Transformer base and medium architectures shows that model size is more important for strong performance than the number of layers. Adding more layers with big

[5] https://github.com/rwth-i6/returnn

|  |  | newsdev2017 (dev) |  |  | newstest2017 |  |  | newstest2018 |  |
|---|---|---|---|---|---|---|---|---|---|
|  | **Systems** | BLEU | TER | cTER | BLEU | TER | cTER | BLEU | cTER |
| 1 | Base | 23.3 | 66.8 | 62.1 | 23.9 | 67.2 | 61.5 | 24.1 | 63.8 |
| 2 | Medium | 24.5 | 65.5 | 61.0 | 24.8 | 65.8 | 60.9 | 25.4 | 63.2 |
| 3 | Large | 24.6 | 65.2 | 60.7 | 25.3 | 65.6 | 60.5 | 26.0 | 62.8 |
| 4 | Ensemble (linear) [1,2,3] | 25.4 | 64.5 | 59.9 | 25.9 | 64.9 | 59.9 | 26.7 | 61.8 |
| 5 | Ensemble (log-linear) [1,2,3] | 25.4 | 64.4 | 60.1 | 26.1 | 64.8 | 59.4 | 26.4 | 62.0 |
| $6^{\dagger}$ | Ensemble (linear) 4 checkpoints of [3] | 24.4 | 65.4 | 60.9 | 25.6 | 65.2 | 60.1 | 26.7 | 62.1 |

Table 6: Results measured in BLEU [%], TER [%] and cTER [%] for Chinese→English. The TER computation on `newstest2018` fails. $^{\dagger}$ indicates the submitted system which is the ensemble of 4 non-converged checkpoints of the large Transformer.

model size and increasing the batch size up to 6500 provides an additional boost of 0.4% BLEU, 0.3% TER and 0.4% cTER on average on all sets (see Row 2 and 3). Furthermore, we try an ensemble of best checkpoints based on BLEU either using various models or using different snapshots of the large Transformer. We use both linear and log-linear ensembling which does not make a difference in terms of BLEU as shown in the Table. Log-linear ensembling is slightly better in terms of TER and is a little bit worse in terms of cTER. We also combine the 4 best checkpoints of the large Transformer shown in Row 6 of Table 6.

## 5 Conclusion

This paper describes the RWTH Aachen University's submission to the WMT 2018 shared news translation task. For German→English our experiments start with a strong baseline which already beats our submission to WMT 2017 by 1.4% BLEU on `newstest2015`. Our final submission is an ensemble of three Transformer models which beats our and the strongest submission of last year by 6.8% BLEU respectively 4.8% BLEU on `newstest2017`. It is ranked first on `newstest2018` by all automatic metrics for this year's news translation task[6]. We suspect that the strength of our systems is especially grounded in the usage of the recently established Transformer architecture, the usage of filtered ParaCrawl in addition to careful experiments on data conditions, the usage of rather big models and large batch sizes, and effective fine-tuning on old test sets.

In English→Turkish task, we show that proper regularization (high dropout rate, weight tying) is crucial for the low-resource setting, yielding a total of up to +7.4% BLEU. Our best system is using 1M sentences synthetic data generated with back-translation (+2.8% BLEU), fine-tuned with test sets of previous year's tasks (+0.7% BLEU), and ensembled over four different training runs (+0.3% BLEU), leading to 18.0% BLEU in `newstest2018`. Note that its cTER is better or comparable to the top-ranked system submissions[7]. In `newstest2017`, our system, even if it is not fine-tuned, outperforms the last year's winning system by +3.6% BLEU.

For our Chinese→English system multiple GPU training that allows for larger models and an increased batch size results in the best preforming single system. A linear ensemble of different Transformer configurations provides 0.7% BLEU, 0.6% TER and 0.8% cTER on average on top of the single best model.

## Acknowledgements

This work has received funding from the European Research Council (ERC) (under the European Union's Horizon 2020 research and innovation programme, grant agreement No 694537, project "SEQCLAS") and the Deutsche Forschungsgemeinschaft (DFG; grant agreement NE 572/8-1, project "CoreTec"). The GPU computing cluster was supported by DFG (Deutsche Forschungsgemeinschaft) under grant INST 222/1168-1 FUGG.
The work reflects only the authors' views and none of the funding agencies is responsible for

[6] `http://matrix.statmt.org/matrix/systems_list/1880`

[7] `http://matrix.statmt.org/matrix/systems_list/1891`

any use that may be made of the information it contains.

## References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*. Version 1.

Parnia Bahar, Jan Rosendahl, Nick Rossenbach, and Hermann Ney. 2017. The RWTH Aachen machine translation systems for IWSLT 2017. In *14th International Workshop on Spoken Language Translation*, pages 29–34.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*. Version 1.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

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.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*. Version 2.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. Version 9.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *CoRR*, abs/1701.06548.

Jan-Thorsten Peter, Eugen Beck, and Hermann Ney. 2018. Sisyphus, a workflow manager designed for machine translation and automatic speech recognition. In *2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

Jan-Thorsten Peter, Andreas Guta, Tamer Alkhouli, Parnia Bahar, Jan Rosendahl, Nick Rossenbach, Miguel Graa, and Ney Hermann. 2017. The RWTH Aachen University English-German and German-English machine translation system for WMT 2017. In *EMNLP 2017 Second Conference on Machine Translation*, Copenhagen, Denmark.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's neural MT systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 389–399.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, volume 2: Shared Task Papers, pages 368–373, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. 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*, pages 223–231, Cambridge, Massachusetts, USA.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open source hierarchical translation, extended with reordering and lexicon models. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 262–270. Association for Computational Linguistics.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. CharacTer: Translation Edit Rate on Character Level. In *The 54th Annual Meeting of the Association for Computational Linguistics - proceedings of the First Conference on Machine Translation (WMT) : August 7-12, 2016, Berlin, Germany : ACL 2016. - Volume 2, Shared Task Papers*, pages 505–510, Stroudsburg, PA. Association for Computational Linguistics.

Albert Zeyer, Tamer Alkhouli, and Hermann Ney. 2018. RETURNN as a generic flexible neural toolkit with application to translation and speech recognition. In *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 128–133.

# 6 Appendix

For our very first experiments with Sockeye a configuration[8] from the Sockeye git repository provided a good starting point.

For our strongest single De→En model (Table 3, Row 7) we ended up with the following options:

```
--num-layers 6:6
--encoder transformer
--decoder transformer
--num-embed 1024:1024
--transformer-model-size 1024
--transformer-feed-forward
   -num-hidden 4096
--transformer-attention-heads 16
--transformer-positional
   -embedding-type fixed
--transformer-preprocess n
--transformer-postprocess dr
--embed-dropout 0:0
--transformer-dropout-prepost 0.1
--transformer-dropout-act 0.1
--transformer-dropout
   -attention 0.1
--label-smoothing 0.1
--learning-rate-reduce-num
   -not-improved 3
--checkpoint-frequency 20000
--batch-type word
--batch-size 5000
--device-ids -4
--grad-accumulation 2    # see*
```

* Note that the `--grad-accumulation` option is introduced by us and is not provided by the official Sockeye version. It refers to the accumulation of gradients, described in Section 4.1, which increases the effective batch-size: In the provided config the effective batch size is 10000.

For our vocabulary sizes (45k and 34k for German and English) the listed configuration results in a Transformer network with 291M trainable parameters.

[8] https://github.com/awslabs/sockeye/blob/arxiv_1217/arxiv/code/transformer/sockeye/train-transformer.sh

# The University of Cambridge's Machine Translation Systems for WMT18

**Felix Stahlberg**[†] and **Adrià de Gispert**[‡†] and **Bill Byrne**[‡†]
[†]Department of Engineering, University of Cambridge, UK
[‡]SDL Research, Cambridge, UK
{fs439,ad465,wjb31}@cam.ac.uk

## Abstract

The University of Cambridge submission to the WMT18 news translation task focuses on the combination of diverse models of translation. We compare recurrent, convolutional, and self-attention-based neural models on German-English, English-German, and Chinese-English. Our final system combines all neural models together with a phrase-based SMT system in an MBR-based scheme. We report small but consistent gains on top of strong Transformer ensembles.

## 1 Introduction

Encoder-decoder networks (Pollack, 1990; Chrisman, 1991; Forcada and Ñeco, 1997; Kalchbrenner and Blunsom, 2013) are the current prevailing architecture for neural machine translation (NMT). Various architectures have been used in the general framework of encoder and decoder networks such as recursive auto-encoders (Pollack, 1990; Socher et al., 2011; Li et al., 2013), (attentional) recurrent models (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Wu et al., 2016; Chen et al., 2018), convolutional models (Kalchbrenner and Blunsom, 2013; Kaiser et al., 2017; Gehring et al., 2017), and, most recently, purely (self-)attention-based models (Vaswani et al., 2017; Ahmed et al., 2017; Shaw et al., 2018). In the spirit of Chen et al. (2018) we devoted our WMT18 submission to exploring the three most commonly used architectures: recurrent, convolutional, and self-attention-based models like the Transformer (Vaswani et al., 2017). Our experiments suggest that self-attention is the superior architecture on the tested language pairs, but it can still benefit from model combination with the other two. We show that using large batch sizes is crucial to Transformer training, and that the delayed SGD updates technique (Saunders et al., 2018) is useful to increase the batch size on limited GPU hardware. Furthermore, we also report gains from MBR-based combination with a phrase-based SMT system. We found this particularly striking as the SMT baselines are often more than 10 BLEU points below our strongest neural models. Our final submission ranks second in terms of BLEU score in the WMT18 evaluation campaign on English-German and German-English, and outperforms all other systems on a variety of linguistic phenomena on German-English (Avramidis et al., 2018).

## 2 System Combination

Stahlberg et al. (2017a) combined SMT and NMT in a hybrid system with a minimum Bayes-risk (MBR) formulation which has been proven useful even for practical industry-level MT (Iglesias et al., 2018). Our system combination scheme is a generalization of this approach to more than two systems. Suppose we want to combine $q$ models $\mathcal{M}_1, \dots, \mathcal{M}_q$. We first divide the models into two groups by selecting a $p$ with $1 \leq p \leq q$. We refer to scores from the first group $\mathcal{M}_1, \dots, \mathcal{M}_p$ as *full posterior* scores and from the second group $\mathcal{M}_{p+1}, \dots, \mathcal{M}_q$ as *MBR-based* scores. Full posterior models contribute to the combined score with their complete posterior of the full translation. In contrast, models in the second group only provide the evidence space for estimating the probability of $n$-grams occurring in the translation. Full-posterior models need to assign scores via the standard left-to-right factorization of neural sequence models:

$$\log P(y_1^T|\mathbf{x}, \mathcal{M}_i) = \sum_{t=1}^{T} \log P(y_t|y_1^{t-1}, \mathbf{x}, \mathcal{M}_i) \tag{1}$$

for a target sentence $\mathbf{y} = y_1^T$ of length $T$ given a source sentence $\mathbf{x}$ for all $i \leq p$. For exam-

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 504–512
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64054

ple, all left-to-right neural models in this work can be used as full posterior models, but the right-to-left models (Sec. 3) and SMT cannot. We combine full-posterior scores log-linearly, and bias the combined score $S(\mathbf{y}|\mathbf{x})$ towards low-risk hypotheses with respect to the MBR-based group as suggested by Stahlberg et al. (2017a, Eq. 4):[1]

$$S(\mathbf{y}|\mathbf{x}) = \sum_{t=1}^{T} \Big( \underbrace{\sum_{i=1}^{p} \lambda_i \log P(y_t|y_1^{t-1}, \mathbf{x}, \mathcal{M}_i)}_{\text{Full posterior}} + \underbrace{\sum_{j=p+1}^{q} \lambda_j \sum_{n=1}^{4} P(y_{t-n}^{t}|\mathbf{x}, \mathcal{M}_j)}_{\text{MBR-based } n\text{-gram scores}} \Big) \quad (2)$$

where $\lambda_1, \dots, \lambda_q$ are interpolation weights. Eq. 2 also describes how to use beam search in this framework as hypotheses can be built up from left to right due to the outer sum over time steps. The MBR-based models contribute via the probability $P(y_{t-n}^{t}|\mathbf{x}, \mathcal{M}_j)$ of an $n$-gram $y_{t-n}^{t}$ given the source sentence $\mathbf{x}$. Posteriors in this form are commonly used for MBR decoding in SMT (Kumar and Byrne, 2004; Tromble et al., 2008), and can be extracted efficiently from translation lattices using counting transducers (Blackwood et al., 2010). For our neural models we run beam search with beam size 15 and compute posteriors over the 15-best list. We smooth all $n$-gram posteriors as suggested by Stahlberg et al. (2017a).

Note that our generalization to more than two systems can still be seen as instance of the original scheme from Stahlberg et al. (2017a) by viewing the first group $\mathcal{M}_1, \dots, \mathcal{M}_p$ as ensemble and the evidence space from the second group $\mathcal{M}_{p+1}, \dots, \mathcal{M}_q$ as mixture model.

The performance of our system combinations depends on the correct calibration of the interpolation weights $\lambda_1, \dots, \lambda_q$. We first tried to use $n$-best or lattice MERT (Och, 2003; Macherey et al., 2008) to find interpolation weights, but these techniques were not effective in our setting, possibly due to the lack of diversity and depth in $n$-best lists from standard beam search. Therefore, we tune on the first best translation using Powell's method (Powell, 1964) with a line search algorithm similar to golden-section search (Kiefer, 1953).

## 3 Right-to-left Translation Models

Standard NMT models generate the translation from left to right on the target side. Recent work has shown that incorporating models which generate the target sentence in reverse order (i.e. from right to left) can improve translation quality (Liu et al., 2016; Li et al., 2017; Sennrich et al., 2017; Hassan et al., 2018). Right-to-left models are often used to rescore $n$-best lists from left-to-right models. However, we could not find improvements from rescoring in our setting. Instead, we extract $n$-gram posteriors from the R2L model, reverse them, and use them for system combination as described in Sec. 2.

## 4 Experimental Setup

### 4.1 Data Selection

We ran language detection (Nakatani, 2010) and gentle length filtering based on the number of characters and words in a sentence on all available monolingual and parallel data in English, German, and Chinese. Due to the high level of noise in the ParaCrawl corpus and its large size compared to the rest of the English-German data we additionally filtered ParaCrawl more aggressively with the following rules:

- No words contain more than 40 characters.
- Sentences must not contain HTML tags.
- The minimum sentence length is 4 words.
- The character ratio between source and target must not exceed 1:3 or 3:1.
- Source and target sentences must be equal after stripping out non-numerical characters.
- Sentences must end with punctuation marks.

This additional filtering reduced the size of ParaCrawl from originally 36M sentences to 19M sentences after language detection, and to 11M sentences after applying the more aggressive rules.

For backtranslation (Sennrich et al., 2016a) we selected 20M sentences from News Crawl 2017. We used a single Transformer (Vaswani et al., 2017) model in Tensor2Tensor's (Vaswani et al., 2018) `transformer_base` configuration

[1] Eq. 2 differs from Eq. 4 of Stahlberg et al. (2017a) in that we do not use a word penalty $\Theta_0$ here, and we do not tune weights for different order $n$-grams separately ($\Theta_1, \dots \Theta_4$). Both did not improve translation quality in our setting.

| Corpus | Over-sampling | #Sentences |
|---|---|---|
| Common Crawl | 2x | 4.43M |
| Europarl v7 | 2x | 3.76M |
| News Commentary v13 | 2x | 0.57M |
| Rapid 2016 | 2x | 2.27M |
| ParaCrawl | 1x | 11.16M |
| Synthetic (news-2017) | 1x | 20.00M |
| **Total** | | **42.19M** |

Table 1: Training data sizes for English-German and German-English after filtering.

| Corpus | Over-sampling | #Sentences |
|---|---|---|
| CWMT - CASIA2015 | 2x | 2.08M |
| CWMT - CASICT2015 | 2x | 3.95M |
| CWMT - Datum2017 | 2x | 1.93M |
| CWMT - NEU2017 | 2x | 3.95M |
| News Commentary v13 | 2x | 0.49M |
| UN v1.0 | 1x | 14.25M |
| Synthetic (news-2017) | 1x | 20.00M |
| **Total** | | **46.66M** |

Table 2: Training data sizes for Chinese-English after filtering.

| Architecture | en-de, de-en | zh-en |
|---|---|---|
| LSTM | 114.2M | 192.7M |
| SliceNet | 27.5M | 86.4M |
| Transformer | 212.8M | 291.4M |
| Relative Transformer | 213.8M | 292.5M |

Table 3: Number of model parameters.

| #Physical GPUs ($g$) | Delay factor ($d$) | #Effective GPUs (g'=gd) | Effective batch size (b'=bg') | BLEU |
|---|---|---|---|---|
| 1 | 1 | 1 | 2,048 | 28.2 |
| 4 | 1 | 4 | 8,192 | 29.5 |
| 4 | 4 | 16 | 32,768 | 30.3 |
| 4 | 16 | 64 | 131,072 | 29.8 |

Table 4: Impact of the effective batch size on Transformer training on en-de news-test2017 after 3,276M training tokens, beam size 4.

for generating the synthetic source sentences. We over-sampled (Sennrich et al., 2017) WMT data by factor 2 except the ParaCrawl data and the UN data on Chinese-English to roughly match the size of the synthetic data. Tabs. 1 and 2 summarize the sizes of our final training corpora.

## 4.2 Preprocessing

We preprocess our English and German data with Moses tokenization, punctuation normalization, and truecasing. On Chinese we first used the WMT `tokenizeChinese.py`[2] script and separated segments of Chinese and Latin text from each other. Then, we removed whitespace between Chinese characters and tokenized Chinese segments with Jieba[3] and the rest with `mteval-v13a.pl`. For our neural models we apply byte-pair encoding (Sennrich et al., 2016b, BPE) with 32K merge operations. We use joint BPE vocabularies on English-German and German-English and separate source/target encodings on Chinese-English.

## 4.3 Model Hyper-Parameters

We use 1024-dimensional embedding and output projection layers in all architectures. The embeddings are shared between encoder and decoder on English-German and German-English, but not on Chinese-English.

**LSTM** For our recurrent models we adapted the TensorFlow seq2seq tutorial code base (Luong et al., 2017) for use inside the Tensor2Tensor library (Vaswani et al., 2018).[4] We roughly followed the UEdin WMT17 submission (Sennrich et al., 2017) and stacked four 1024-dimensional LSTM layers with layer normalization (Ba et al., 2016) and residual connections in both the decoder and bidirectional encoder. We equipped the decoder network with Bahdanau-style (Bahdanau et al., 2015) attention (`normed_bahdanau`).

**SliceNet** The convolutional model of Kaiser et al. (2017) called SliceNet is implemented in Tensor2Tensor. We use the standard configuration `slicenet_1` of four hidden layers with layer normalization.

**Transformer** We compare two Transformer variants available in Tensor2Tensor: the original Transformer (Vaswani et al., 2017) (`transformer_big` setup) and the Transformer of Shaw et al. (2018) with relative positional embeddings (`transformer_relative_big` setup). Both use 16-head dot-product attention and six 1024-dimensional encoder and decoder layers.

The number of training parameters of our neural models is summarized in Tab. 3.

[2] `http://www.statmt.org/wmt17/tokenizeChinese.py`

[3] `https://github.com/fxsjy/jieba`

[4] `https://github.com/fstahlberg/tensor2tensor-usr`

| Architecture | #Effective GPUs | Batch size | #SGD updates | #Training tokens |
|---|---|---|---|---|
| LSTM | 8 | 4,096 | 45K | 1,475M |
| SliceNet | 4 | 2,048 | 800K | 6,554M |
| R2L Transformer | 16 | 2,048 | 200K | 6,554M |
| Transformer | 16 | 2,048 | 250K | 8,192M |
| Relative Transformer | 16 | 2,048 | 250K | 8,192M |

Table 5: Training setups for our neural models on all language pairs.

### 4.4 Training

We train vanilla phrase-based SMT systems[5] and extract 1000-best lists of unique translations candidates, from which $n$-gram posteriors are calculated.

All neural models were trained with the Adam optimizer (Kingma and Ba, 2015), dropout (Srivastava et al., 2014), and label smoothing (Szegedy et al., 2016) using the Tensor2Tensor (Vaswani et al., 2018) library. We decode with the average of the last 40 checkpoints (Junczys-Dowmunt et al., 2016a).

We make extensive use of the delayed SGD updates technique we already applied successfully to syntax-based NMT (Saunders et al., 2018). Delaying SGD updates allows to arbitrarily choose the effective batch size even on limited GPU hardware. Large batch training has received some attention in recent research (Smith et al., 2017; Neishi et al., 2017) and has been shown particularly useful for training the Transformer architecture with the Tensor2Tensor framework (Popel and Bojar, 2018). We support these findings in Tab. 4.[6] Our technical infrastructure[7] allows us to train on four P100 GPUs simultaneously, which limits the number of physical GPUs to $g = 4$ and the batch size[8] to $b = 2048$ due to the GPU memory. Thus, the maximum possible effective batch size without delaying SGD updates is $b' = 8192$. Training with delay factor $d$ accumulates gradients over $d$ batches and applies the optimizer update rule on the accumulated gradients. This allows us to scale up the effective number of GPUs to 16 and improve the BLEU score significantly (29.5 vs. 30.3). Note that training regimens are equivalent if their effective batch size is the same, ie. training on 4 physical GPUs with $d = 4$ is mathematically equivalent to training on 16 GPUs without delaying SGD updates. Tab. 5 lists our training setups for the neural architectures used in this work. These training hyper-parameters were chosen empirically. Particularly, we did not find improvements by increasing the number of effective GPUs for SliceNet or longer LSTM training.

We use *news-test2017* as development set on all language pairs to tune the model interpolation weights $\lambda$ (Eq. 2) and the scaling factor for length normalization.

### 4.5 Decoding

We use the `beam` search strategy with beam size 8 of the SGNMT decoder (Stahlberg et al., 2017b, 2018) in all our experiments. We apply length normalization (Bahdanau et al., 2015) on German-English and Chinese-English but not on English-German. As outlined in Sec. 2 we either use full posteriors or MBR-style $n$-gram posteriors from our individual models. SMT $n$-gram scores are extracted as described by Blackwood et al. (2010) using HiFST's `lmbr` tool. We use SGNMT's `ngram` output format to extract $n$-gram scores from our neural models.

## 5 Results

On English-German and German-English *news-test2014* we compute cased BLEU scores with Moses' `multi-bleu.pl` script on tokenized output to be comparable with prior work (Wu et al., 2016; Kaiser et al., 2017; Gehring et al., 2017; Vaswani et al., 2017; Chen et al., 2018). On all other test sets we use `mteval-v13a.pl` to be comparable to the official cased WMT scores.[9]

First, we will discuss our experiments with a single architecture, i.e. single systems and ensembles of two systems with the same architecture. Tab. 6 compares the architectures on all test sets. PBMT as a single system is clearly inferior to all neural systems. Ensembling neural systems helps for all architectures across the board. LSTM

[5] Excluding the UN corpus and the backtranslated data.

[6] We had to reduce the learning rate for $g' = 1$ to avoid training divergence.

[7] `http://www.hpc.cam.ac.uk/`

[8] We follow Vaswani et al. (2017, 2018) and specify the batch size in terms of number of source and target tokens in a batch, not the number of sentences.

[9] `http://matrix.statmt.org/`

| Architecture | #Systems | English-German | | | | German-English | | | | Chinese-English | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | test14 | test15 | test16 | test17 | test14 | test15 | test16 | test17 | dev17 | test17 |
| PBMT | 1 | 19.6 | 20.9 | 25.6 | 20.0 | 22.5 | 27.2 | 32.6 | 28.2 | 14.2 | 15.8 |
| LSTM | 1 | 27.1 | 28.8 | 34.6 | 28.0 | 33.8 | 33.3 | 40.7 | 34.8 | 21.8 | 22.7 |
| | 2 | 28.2 | 29.6 | 35.5 | 28.5 | 34.6 | 34.0 | 41.4 | 35.3 | 22.7 | 23.6 |
| SliceNet | 1 | 26.8 | 28.9 | 33.6 | 27.6 | 32.6 | 32.3 | 39.8 | 33.7 | 21.4 | 22.5 |
| | 2 | 27.2 | 29.6 | 34.6 | 28.3 | 33.2 | 32.9 | 40.8 | 34.3 | 21.8 | 23.4 |
| R2L Trans. | 1 | 30.3 | 31.5 | 36.3 | 30.2 | 36.5 | 35.5 | 43.5 | 37.2 | 24.5 | 24.9 |
| Transformer | 1 | 30.7 | 31.9 | 36.6 | 30.5 | 36.7 | 36.2 | 43.7 | 37.9 | 24.9 | 25.6 |
| | 2 | 31.1 | 31.8 | 37.2 | 31.0 | 36.9 | 36.4 | 44.0 | 38.1 | 26.2 | 26.2 |
| Rel. Trans. | 1 | 31.2 | 31.9 | 37.0 | 31.1 | 37.0 | 36.3 | 44.1 | 38.1 | 24.9 | 25.8 |
| | 2 | 31.4 | 32.3 | 37.7 | 31.2 | 37.2 | 36.5 | 44.1 | 38.4 | 25.1 | 26.4 |

Table 6: Single architecture results on all language pairs for single systems and 2-ensembles.

| | Full posterior | | | | | MBR-based $n$-gram scores | | | | BLEU (test2017) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PBMT | LSTM* | SliceNet* | Trans. | Rel. Trans. | PBMT | LSTM* | SliceNet* | R2L Trans. | en-de | de-en | zh-en |
| 1 | ✓ | | | | | | | | | 20.0 | 28.2 | 15.8 |
| 2 | | ✓ | | | | | | | | 28.5 | 35.3 | 23.6 |
| 3 | | | ✓ | | | | | | | 28.3 | 34.3 | 23.4 |
| 4 | | | | ✓ | | | | | | 30.5 | 37.9 | 25.6 |
| 5 | | | | | ✓ | | | | | 31.1 | 38.1 | 25.8 |
| 6 | | | | ✓ | ✓ | | | | | 31.3 | 38.2 | 26.4 |
| 7 | | ✓ | ✓ | ✓ | ✓ | | | | | 31.3 | 38.2 | 26.4 |
| 8 | | | | ✓ | ✓ | | ✓ | ✓ | | 31.4 | 38.2 | 26.6 |
| 9 | | | | ✓ | ✓ | | ✓ | ✓ | ✓ | 31.4 | 38.3 | 26.8 |
| 10 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 31.7 | 38.7 | 27.1 |

Table 7: Model combination with ensembling and MBR.Model scores are weighted with MERT and combined (log-)linearly as described in Sec. 2. *: The LSTM and SliceNet models are 2-ensembles.

is usually slightly better than the convolutional SliceNet, but is much slower to train and decode (cf. Tab. 3). Note that our LSTM 2-ensemble is on par with the best BLEU score in WMT17 (Sennrich et al., 2017), which was also based on recurrent models. Transformer architectures outperform LSTMs and SliceNets on all test sets. The right-to-left Transformer is usually slightly worse, the Transformer with relative positioning slightly better than the standard Transformer setup.

Tab. 7 summarizes our system combination results with multiple architectures. Adding LSTM and SliceNet as full-posterior models to an ensemble of a Transformer and a Relative Transformer does not improve the BLEU score (rows 6 vs. 7). We see very slight improvements when we use these models to extract $n$-gram scores instead (rows 6 vs. 8). We report further gains by using MBR-based $n$-gram scores from the right-to-left Transformer and the PBMT system. The improvements from adding PBMT are rather small, but we still found them surprising given that the PBMT baseline is usally more than 10 BLEU points worse than our best single neural model. We list the performance of our submitted systems on all test sets in Tab. 8.

| Direction | Test set | BLEU |
|---|---|---|
| English-German | news-test14 | 31.6 |
| | news-test15 | 32.6 |
| | news-test16 | 38.5 |
| | news-test17 | 31.7 |
| | news-test18 | 46.6 |
| German-English | news-test14 | 36.8 |
| | news-test15 | 36.5 |
| | news-test16 | 45.1 |
| | news-test17 | 38.7 |
| | news-test18 | 48.0 |
| Chinese-English | news-dev17 | 25.7 |
| | news-test17 | 27.1 |
| | news-test18 | 27.7 |

Table 8: BLEU scores of the submitted systems (row 10 in Tab. 7).

## 6 Related Work

There is a large body of research comparing NMT and SMT (Schnober et al., 2016; Toral and Sánchez-Cartagena, 2017; Koehn and Knowles, 2017; Menacer et al., 2017; Dowling et al., 2018; Bentivogli et al., 2016, 2018). Most studies have found superior overall translation quality of NMT models in most settings, but complementary strengths of both paradigms. Therefore, the literature about hybrid NMT-SMT systems is also vast, ranging from rescoring and reranking methods (Neubig et al., 2015; Stahlberg et al., 2016; Khayrallah et al., 2017; Grundkiewicz and Junczys-Dowmunt, 2018; Avramidis et al., 2016; Marie and Fujita, 2018), MBR-based formalisms (Stahlberg et al., 2017a, 2018; Iglesias et al., 2018), NMT assisting SMT (Junczys-Dowmunt et al., 2016b; Du and Way, 2017), and SMT assisting NMT (Niehues et al., 2016; He et al., 2016; Long et al., 2016; Wang et al., 2017; Dahlmann et al., 2017; Zhou et al., 2017). We confirm the potential of hybrid systems by reporting gains on top of very strong neural ensembles.

Ensembling is a well-known technique in NMT to improve system performance. However, ensembles usually consist of multiple models of the same architecture. In this paper, we compare and combine three very different architectures (recurrent, convolutional, and self-attention based) in two different ways (full posterior and MBR-based), and find that combination with MBR-based $n$-gram scores is superior.

## 7 Conclusion

We have described our WMT18 submission, which achieves very competitive BLEU scores on all three language pairs (English-German, German-English, and Chinese-English) and significantly higher accuracies in a variety of linguistic phenomena compared to other submissions (Avramidis et al., 2018). Our system combines three different neural architecture with a traditional PBMT system. We showed that our MBR-based scheme is effective to combine these diverse models of translation, and that adding the PBMT system to the mix of neural models still yields gains although it is much worse as stand-alone system.

## Acknowledgments

This work was supported in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC grant EP/L027623/1).

## References

Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. 2017. Weighted transformer network for machine translation. *arXiv preprint arXiv:1711.02132*.

Eleftherios Avramidis, Vivien Macketanz, Aljoscha Burchardt, Jindrich Helcl, and Hans Uszkoreit. 2016. Deeper machine translation and evaluation for German. In *Proceedings of the 2nd Deep Machine Translation Workshop*, pages 29–38. ÚFAL MFF UK.

Eleftherios Avramidis et al. 2018. Fine-grained evaluation of German-English machine translation based on a test suite. In *Proceedings of the Third Conference on Machine Translation, Volume 3*, Brussels, Belgium. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267. Association for Computational Linguistics.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2018. Neural versus phrase-based MT quality: An in-depth analysis on English–German and English–French. *Computer Speech & Language*, 49:52–70.

Graeme Blackwood, Adrià Gispert, and William Byrne. 2010. Efficient path counting transducers for minimum Bayes-risk decoding of statistical machine translation lattices. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 27–32. Association for Computational Linguistics.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*.

Lonnie Chrisman. 1991. Learning recursive distributed representations for holistic computation. *Connection Science*, 3(4):345–366.

Leonard Dahlmann, Evgeny Matusov, Pavel Petrushkov, and Shahram Khadivi. 2017. Neural machine translation leveraging phrase-based models in a hybrid search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1411–1420. Association for Computational Linguistics.

Meghan Dowling, Teresa Lynn, Alberto Poncelas, and Andy Way. 2018. SMT versus NMT: Preliminary comparisons for Irish. *Technologies for MT of Low Resource Languages (LoResMT 2018)*, page 12.

Jinhua Du and Andy Way. 2017. Neural pre-translation for hybrid machine translation. *In Proceedings of MT Summit XVI*, 1:27–40.

Mikel L. Forcada and Ramón P. Ñeco. 1997. Recursive hetero-associative memories for translation. In *Biological and Artificial Computation: From Neuroscience to Technology*, pages 453–462, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *ArXiv e-prints*.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic Chinese to English news translation. *arXiv preprint arXiv:1803.05567*.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with SMT features. In *AAAI*, pages 151–157.

Gonzalo Iglesias, William Tambellini, Adrià de Gispert, Eva Hasler, and Bill Byrne. 2018. Accelerating NMT batched beam decoding with LMBR posteriors for deployment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016a. Is neural machine translation ready for deployment? A case study on 30 translation directions. In *Proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*, Seattle, WA.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Rico Sennrich. 2016b. The AMU-UEDIN submission to the WMT16 news translation task: Attention-based NMT models as feature functions in phrase-based smt. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 319–325. Association for Computational Linguistics.

Lukasz Kaiser, Aidan N Gomez, and Francois Chollet. 2017. Depthwise separable convolutions for neural machine translation. *arXiv preprint arXiv:1706.03059*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Huda Khayrallah, Gaurav Kumar, Kevin Duh, Matt Post, and Philipp Koehn. 2017. Neural lattice search for domain adaptation in machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 20–25. Asian Federation of Natural Language Processing.

Jack Kiefer. 1953. Sequential minimax search for a maximum. *Proceedings of the American mathematical society*, 4(3):502–506.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*.

Aodong Li, Shiyue Zhang, Dong Wang, and Thomas Fang Zheng. 2017. Enhanced neural machine translation by learning from draft. In *Proceedings of APSIPA Annual Summit and Conference*, volume 2017, pages 12–15.

Peng Li, Yang Liu, and Maosong Sun. 2013. Recursive autoencoders for ITG-based translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 567–577, Seattle, Washington, USA. Association for Computational Linguistics.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416. Association for Computational Linguistics.

Zi Long, Takehito Utsuro, Tomoharu Mitsuhashi, and Mikio Yamamoto. 2016. Translation of patent sentences with a large vocabulary of technical terms using neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 47–57. The COLING 2016 Organizing Committee.

Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. 2017. Neural machine translation (seq2seq) tutorial. *https://github.com/tensorflow/nmt*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 725–734. Association for Computational Linguistics.

Benjamin Marie and Atsushi Fujita. 2018. A smorgasbord of features to combine phrase -based and neural machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, Boston, US.

Mohamed-Amine Menacer, David Langlois, Odile Mella, Dominique Fohr, Denis Jouvet, and Kamel Smaïli. 2017. Is statistical machine translation approach dead? In *ICNLSSP 2017-International Conference on Natural Language, Signal and Speech Processing*.

Shuyo Nakatani. 2010. Language detection library for Java.

Masato Neishi, Jin Sakuma, Satoshi Tohda, Shonosuke Ishiwatari, Naoki Yoshinaga, and Masashi Toyoda. 2017. A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 99–109.

Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. In *WAT*, Kyoto, Japan.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1828–1836. The COLING 2016 Organizing Committee.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

Jordan B. Pollack. 1990. Recursive distributed representations. *Artificial Intelligence*, 46(1):77 – 105.

Martin Popel and Ondřej Bojar. 2018. Training tips for the Transformer model. *arXiv preprint arXiv:1804.00247*.

Michael JD Powell. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162.

Danielle Saunders, Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2018. Multi-representation ensembles and delayed SGD updates improve syntax-based NMT. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. To appear.

Carsten Schnober, Steffen Eger, Erik-Lân Do Dinh, and Iryna Gurevych. 2016. Still not there? Comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1703–1714. The COLING 2016 Organizing Committee.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's neural MT systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Samuel L Smith, Pieter-Jan Kindermans, and Quoc V Le. 2017. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Felix Stahlberg, Adrià de Gispert, Eva Hasler, and Bill Byrne. 2017a. Neural machine translation by minimising the Bayes-risk with respect to syntactic translation lattices. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 362–368. Association for Computational Linguistics.

Felix Stahlberg, Eva Hasler, Danielle Saunders, and Bill Byrne. 2017b. SGNMT – A flexible NMT decoding platform for quick prototyping of new models and search strategies. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 25–30. Association for Computational Linguistics.

Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. 2016. Syntactically guided neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 299–305. Association for Computational Linguistics.

Felix Stahlberg, Danielle Saunders, Gonzalo Iglesias, and Bill Byrne. 2018. Why not be versatile? Applications of the SGNMT decoder for machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, Boston, US.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Antonio Toral and Víctor M. Sánchez-Cartagena. 2017. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1063–1073. Association for Computational Linguistics.

Roy Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 620–629. Association for Computational Linguistics.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2tensor for neural machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, Boston, US.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. 2017. Neural machine translation advised by statistical machine translation. In *AAAI*, pages 3330–3336.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Long Zhou, Wenpeng Hu, Jiajun Zhang, and Chengqing Zong. 2017. Neural system combination for machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 378–384. Association for Computational Linguistics.

# The LMU Munich Unsupervised Machine Translation Systems

**Dario Stojanovski, Viktor Hangya, Matthias Huck** and **Alexander Fraser**
Center for Information and Language Processing
LMU Munich
{stojanovski,hangyav,mhuck,fraser}@cis.lmu.de

## Abstract

We describe LMU Munich's unsupervised machine translation systems for English↔German translation. These systems were used to participate in the WMT18 news translation shared task and more specifically, for the unsupervised learning sub-track. The systems are trained on English and German monolingual data only and exploit and combine previously proposed techniques such as using word-by-word translated data based on bilingual word embeddings, denoising and on-the-fly backtranslation.

## 1 Introduction

The LMU Munich's Center for Information and Language Processing participated in the WMT 2018 news translation shared task for English↔German translation. Specifically, we participated in the unsupervised learning task which focuses on training MT models without access to any parallel data. The team has a strong track record at previous WMT shared tasks (Bojar et al., 2017, 2016, 2015, 2014, 2013) working on SMT systems (Cap et al., 2014, 2015; Weller et al., 2013; Sajjad et al., 2013; Huck et al., 2016; Peter et al., 2016; Tamchyna et al., 2016) and proposed a top scoring linguistically informed neural machine translation system (Huck et al., 2017) based on human evaluation at WMT17.

Neural machine translation (NMT) is state-of-the-art in automatic translation. Attention-based neural sequence-to-sequence models (Bahdanau et al., 2015) have been established as the basis for most recent work in MT and furthermore, have been used to obtain best scoring systems at WMT in recent years (Bojar et al., 2017, 2016). Previous work and the best scoring systems at WMT also showed that NMT can be scaled to millions of sentence pairs and even achieve human parity (Hassan et al., 2018). However, this comes under the caveat that we have access to a large amount of human-translated parallel data. Koehn and Knowles (2017) showed that NMT models cannot be properly trained under low resource conditions and are still behind phrase-based models. In extremely low resource scenarios, NMT fails completely which is a big obstacle if we want to enable automatic translation over a variety of languages. This motivates the unsupervised learning task at WMT this year. The task is run for three language pairs, but we only focus on English↔German translation. Although this language pair has an abundance of parallel data, we are constrained to only using monolingual data provided for the WMT18 news translation task, excluding Europarl and News Commentary because of content overlap.

The systems we use for our submissions are based on the recently proposed techniques for unsupervised machine translation by several studies (Artetxe et al., 2018; Lample et al., 2018a,b). The phrase-based unsupervised system uses bilingual word embeddings (BWEs) to create an initial phrase table and also utilizes a target-side $n$-gram language model. The backbone of the unsupervised NMT methods is denoising and on-the-fly backtranslation which enable a standard NMT architecture to be trained by only leveraging monolingual data. The model for our submission is mostly based on the work of Lample et al. (2018b). Additionally, we explore how word-by-word translated data based on BWEs can be utilized to improve the initial training and experiment with different ways of producing these translations. We also show that disabling denoising in the last stages of learning can provide for further improvements. We refer the reader to Huck et al. (2018) for our supervised systems for news and biomedical translation.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 513–521
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64055

The remainder of the paper outlines the methods we used for generating BWEs, training a phrase-based and neural unsupervised machine translations systems. Moreover, it presents the obtained results as well as translation examples showcasing some of the strong and weak points of the NMT system.

## 2 Bilingual Word Embeddings

Both our phrase-based and neural unsupervised machine translation systems are based on bilingual word embeddings which represent source and target language words in a shared vector space. Recently, Conneau et al. (2017) showed that good quality bilingual embeddings can be produced by training monolingual models for both source and target languages and mapping them to a shared space without any bilingual signal. We follow this approach and use bilingual word embeddings, trained in an unsupervised fashion, to jump-start both of our systems.

As our baseline system we produce **word-by-word** translations relying only on the embeddings. For each word $w_s$ in the source sentence we induce its translation:

$$tr_{wbw}(w_s) = \arg\max_{w \in V_t} cos(e(w_s), e(w))$$

where $e(w)$ is the vector representation of word $w$, $cos(x, y)$ is the cosine similarity of two vectors and $V_t$ is the target vocabulary.

One problem with the approach arises when translating German compound words which are combinations of two or more words that function as a single unit of meaning. In most of the cases, these words should be translated into multiple English words which causes errors when translating them word by word. The issue is also present when translating from English to German since multiple words should be transformed into one unit. To overcome this issue we experimented with **bigrams** in addition to unigrams. We tried a simple idea, namely, we looked for frequent bigrams in the English side of both the monolingual input data and the test set. We replaced bigrams with their concatenated forms in the sentences and also kept the original sentence. By training bilingual word embeddings on this data we automatically allow the word-by-word algorithm to translate compound words to bigrams and vice-versa.

To further improve the quality of our algorithm, we exploited **orthographic similarity** of words. Braune et al. (2018) showed that the performance of inducing word translations can be significantly improved using orthography. Following the approach there, we obtained improvements, especially when translating named entities, by using the following word translation function:

$$tr_{orth}(w_s) = \arg\max_{w \in V_t} \max \begin{pmatrix} cos(e(w_s), e(w)), \\ \lambda * orth(w_s, w)) \end{pmatrix}$$

where $\lambda$ is a weighting constant and $orth(w_1, w_2)$ is the normalized Levenshtein distance of words $w_1$ and $w_2$.

As a contrastive set of experiments we added light supervision during the training of bilingual word embeddings in order to show performance differences compared to the fully unsupervised setup. To map monolingual spaces we used orthogonal mapping (Xing et al., 2015) with a seed lexicon of of 5000 word pairs, which was used as a baseline in (Conneau et al., 2017) as well.

### 2.1 Technical Details

To train monolingual word embeddings we used *fasttext* (Bojanowski et al., 2017) which employs subword information for better quality representations. We used 512 dimensional embeddings and default values for the rest of the parameters. For both unsupervised and lightly supervised mapping we used *MUSE* (Conneau et al., 2017) with default parameters. We fine-tuned $\lambda$ on the test set of *WMT 2017* and used the method of (Mikolov et al., 2013) to mine frequent bigrams.

## 3 Unsupervised Phrase-based Translation

We have investigated unsupervised phrase-based translation (PBT). The results have been worse than with the neural model in our experiments. In this section, we therefore only give a short outline of the methods which we have explored in that area.

By means of a straightforward format conversion of the BWE lexicon, we can create a word-based "phrase table" that can be loaded into the Moses decoder (Koehn et al., 2007). The cosine similarities from the BWE model become feature scores in the phrase table. Note that we refrained from normalizing the cosine similarities, but wrote their values directly to the table.

Using Moses for decoding carries the advantage that an $n$-gram language model can be integrated without any implementation effort. Once we have added a language model, we can also activate reordering. A distance-based distortion cost may then be added as a further feature.

An obvious difficulty is how to choose the weights for the features. If we assume that a small amount of bitext is actually available (say, a few hundred sentence pairs), then we can tune the weights with MERT or MIRA. We did the latter and built tuned unsupervised phrase-based systems in the outlined way for both translation directions.

With this initial system, we created synthetic training data. We translated around 50 M monolingual sentences from German to English. Not only the translations, but also the decoding word alignments were stored. Next, phrases can be extracted from the synthetic parallel corpus. We can use this new phrase table in the Moses decoder to build a better English→German unsupervised phrase-based system. The feature weights can be tuned again with MERT/MIRA. Word penalty and phrase penalty become useful with the phrase table from synthetic data. The new phrase table contains phrases of different lengths, not only words (or word bigrams).

We trained an English→German unsupervised phrase-based system according to the pipeline that we just described. Its output was uploaded as a contrastive submission, but we decided to not earmark it for manual evaluation.

## 4 Unsupervised Neural Translation

The system we used in this work builds on previous work on unsupervised neural machine translation (Artetxe et al., 2018; Lample et al., 2018a,b). We mostly make use of the techniques suggested in Lample et al. (2018b).

Before training the unsupervised NMT system proposed in Lample et al. (2018b), it is important to properly initialize certain key components which are otherwise randomly initialized. For that purpose, they propose to initialize the encoder and decoder embeddings with BPE-level embeddings trained using *fasttext* (Bojanowski et al., 2017). The BPE splitting is computed jointly on the German and English monolingual data. Given that these two languages are related and share surface forms, this technique is a reasonable choice.

The model proposed in Lample et al. (2018b) consists of two main components, a denoising and a translation component. The denoising part acts as a language model and is trained to produce fluent output in a given language based on a noisy version of the input. We follow the implementation of Artetxe et al. (2018) where the noisy version of the input sentence is obtained by making random swaps of contiguous words. Denoising helps to produce fluent output, but it's also used to enable reordering, and insertions and deletions of words. This is necessary since the model initially tends to do word-by-word translations while in German and English the word order is different.

The translation component works in a traditional way. However, given that the model doesn't have access to parallel data, it needs to make use of on-the-fly backtranslation. During training, the same model is used to backtranslate a sentence from the monolingual data and this pair of backtranslated sample/gold standard sample is used to train the model in a traditional fashion.

In order to enable for the denoising, or language model effects to be transferred to the translation component, many parameters in the model are shared. The encoder is shared for German and English. This forces the model to produce a language-agnostic representation of the input sentence. It also enables for the decoder and the attention mechanism to be shared across both languages. Although the decoder is shared, a language identifier token is added at the beginning of each sentence only on the target side. In our experiments, we observed problems if we try to share the softmax layer, because the output tended to be a mixture of both German and English.

In the model used for our final submission, we use all of the outlined techniques from Lample et al. (2018b). However, we used additional data in the initial learning procedure and modified the training curricula in order to improve performance. In our experiments, we observed some initial training difficulties. As a result, in order to facilitate faster and easier learning, we make use of word-by-word translated synthetic parallel data, in addition to initializing the encoder and decoder embeddings. In our model, the training consists of alternative batches of monolingual data used for denoising and backtranslation and the word-by-word translated synthetic data. The word-by-word translations are obtained as described in Section 2.

We also apply BPE splitting on this data before using it in training.

After a certain number of iterations, we stop with the training of the initial model and "unplug" two components of the previous training procedure. Namely, we remove the word-by-word translated data since this is useful to jump-start the learning, but later presumably will impede learning more nuanced translations. We also observe better results if we disable the denoising component and continue the training by only doing on-the-fly backtranslation. This improved results on both translation directions by more than 1 BLEU (Papineni et al., 2002). However, in subsequent experiments we observed that this can also lead to unstable learning and decrease the performance since bad translation decisions can be reinforced. As a result, the final training procedure should be carefully controlled.

As mentioned in Section 2, the model has problems translating named entities. This stems from the fact that it is dependent on BWEs, where two different named entities often mistakenly have similar representations, causing confusion. Following the improvements the word-by-word translation obtained by using orthographic similarity, we also try training a model with word-by-word translated data utilizing this similarity. We also use word-by-word translated data obtained by using bigrams and orthographic similarity.

## 5 Empirical Evaluation

The models in this work are trained on German and English NewsCrawl articles from 2007 to 2017. Since the total size of this data is very large, we randomly sampled 4M sentences for each language. Moreover, we study if there is any noticeable effect if we only utilize more recent data. As a result, we sampled 4M samples from NewsCrawl 2017 and report results with this dataset as well.

The datasets are tokenized and truecased with the standard scripts from the Moses toolkit (Koehn et al., 2007). When training the truecase models, we actually use all of the available NewsCrawl data, rather than our subsample. We also use BPE splitting. The BPE segmentation is computed jointly on all the NewsCrawl data available for both languages. Then, all sentences with more than 50 tokens are discarded. The NewsCrawl data is also used to train the BPE-level embeddings.

We implement our neural system on top of the code made available by Artetxe et al. (2018). The model is an attention-based encoder-decoder NMT with 2-layer GRU encoder and decoder. The number of hidden units is 600. We set the learning rate to 0.0002 and dropout in the encoder and decoder to 0.3. We checkpoint the model each 10K updates. The batch size is 32.

| | BWE unsupervised | |
|---|---|---|
| | de-en | en-de |
| wbw | 11.50 | 6.94 |
| wbw+bigram | 11.77 | 6.75 |
| wbw+orth | 12.37 | 7.92 |
| wbw+orth+bigram | 12.58 | 7.64 |
| | BWE lightly supervised | |
| | de-en | en-de |
| wbw | 10.99 | 7.28 |
| wbw+bigram | 11.28 | 7.08 |
| wbw+orth | 11.70 | 8.24 |
| wbw+orth+bigram | 11.98 | 7.93 |

Table 1: Baseline results (BLEU) with word-by-word translation on newstest2018. We indicate the use of bigrams and orthographic similarity with *bigram* and *orth* respectively.

### 5.1 BWE Baseline Experiments

We present our word-by-word translation baseline results in Table 1. Using bigrams on the English side helped for de-en but not for en-de. By analyzing translations we can conclude that 1) German compound words are correctly translated to multiple words in many cases and 2) the drop of en-de direction is caused by incorrectly translating bigrams, that are non-compounds on the target side, to one token units. On the other hand, using orthographic information gave significant improvements in both directions. The technique alone provided for improved translation of named entities without the use of a costly NER system. We got our best results by combining bigrams and orthographic similarity for German→English.

Comparing the results with the unsupervised and lightly supervised mapping it can be seen that the two systems are on par in performance, the former results higher BLEU points in case of de-en but lower for en-de. Our conjecture is that the multiple translations of the source words in the used lexicon helped tackle the morphological richness of the German language on the target side while it was not helpful otherwise.

## 5.2 Unsupervised PBT Results

The top half of Table 2 reports the translation quality that we achieved with the phrase-based unsupervised approach (cf. Section 3), measured in case-sensitive BLEU. Our test set for these experiments is newstest2017 (whereas the BLEU scores in Table 1 are on newstest2018). The experiment in the first line of Table 2 is conceptually equivalent to the unsupervised "wbw" experiment from Table 1. We use the Moses decoder to perform monotonic word-by-word translation without a language model (LM) or any other feature functions except for the single translation model (TM) score that we obtain from the cosine similarities. If we add a 4-gram LM and heuristically weight the LM feature function with a scaling factor of 0.1 and the TM with 0.9 (second line in Table 2), the translation quality improves by more than 2.5 BLEU points in both of the two translation directions. By using a small parallel development set (newstest2016) to tune the two weights with MIRA (Cherry and Foster, 2012) (third line), we barely improve over our guessed scaling factors of 0.1 for the LM and 0.9 for the TM. Optimized scaling factors are however more relevant when we allow for reordering (fourth line), since we then activate a third feature function, namely a distance-based distortion cost. This adds another scaling factor, and a good informed guess of reasonable values for three weights becomes increasingly difficult. Activated reordering with tuned weights boosts our translation quality further.

We can go beyond simple word-by-word translation if we add our BWE bigrams to the TM, thus also enabling 1:2, 2:1, and 2:2 translation by means of new phrase table entries. Reordering and the 4-gram LM are kept active in the new configuration. But to give the system control over the lengths of the hypothesis translations (which now can differ from the input sentence lengths), we also activate the word penalty and phrase penalty feature functions, and we include three more binary indicator features for table entries that are 1:2, 2:1, and 2:2, respectively. The scaling factors are optimized on newstest2016 again. With bigrams, we observe higher translation quality in the German→English translation direction, but not in the English→German direction (fifth line in Table 2). This is consistent with what we noted above (cf. Table 1).

Finally, we created 50 M synthetic sentence pairs from German monolingual data with our best German→English phrase-based unsupervised system. With a phrase table extracted from the synthetic data, we achieve our best phrase-based unsupervised translation result in the English→German translation direction (sixth line).[1]

|  | unsup. PBT |  |
|---|---|---|
|  | de-en | en-de |
| wbw (Moses decoder) | 7.92 | 4.49 |
| + 4-gram LM (weighted 0.1) | 10.52 | 7.21 |
| + tuned weights | 10.73 | 7.20 |
| + reordering | 11.47 | 7.66 |
| + bigram | 12.44 | 7.61 |
| synthetic data | – | 10.66 |
|  | unsup. NMT |  |
|  | de-en | en-de |
| baseline | 13.77 | 10.45 |
| fine-tune w/o denoising | 15.03 | 12.08 |
| w/ orth | 16.06 | 12.38 |
| w/ orth + bigram | 16.98 | 13.13 |
| NewsCrawl 2017 | 16.42 | 12.46 |

Table 2: BLEU scores with the unsupervised systems on newstest2017.

## 5.3 Unsupervised NMT Results

We show the results from our unsupervised neural systems (cf. Section 4) in the bottom half of Table 2. The translation quality still lags behind supervised translation systems. Only one other team (RWTH Aachen University) competed in the WMT18 unsupervised learning sub-track, and the performance of their unsupervised systems is roughly comparable to our submissions.

Our final submission system was trained on a subsample of NewsCrawl from 2007 to 2017. We did not include any of the orthographic similarity or bigram word-by-word translated data. The model selection was done based on the newstest2017 test set and we use the same model checkpoint for both translation directions. For the final submission model, we removed the word-by-word translated data after 6K iterations and subsequently trained the model for a total of 300K iterations. This model was able to obtain 13.77 on the de-en and 10.45 on en-de translation task. Subsequently, we disabled denoising and contin-

[1] In consideration of the computational cost, we decided to try synthetic data in only one of the two translation directions.

ued the training just with on-the-fly backtranslation which managed to provide for further gains of 1.26 for de-en and 1.63 for en-de. In subsequent experiments we observed that removing the word-by-word translated data does not change the performance and for the contrastive experiments, for simplicity, we remove it at the same time as disabling denoising.

Our contrastive experiments show that the choice of data can have some effect on the translation performance. Training a model on a subsample of NewsCrawl 2017, showed to be more beneficial. Using more recent data can provide for better correlation between the training and test sets. However, it is difficult to pinpoint whether this is because of better general content overlap or because of the recency of the data.

In the word-by-word translations, the use of orthographic similarity proved to be very helpful. Some of those effects are transfered when we use that data in the neural system. For de-en it provided for an improvement of 1.03 BLEU, while for en-de only 0.30 BLEU.

Adding bigrams did not provide for consistent improvements in the word-by-word translations. However, the neural system managed to make use of these translations better, most likely from the additional reordering that is contained in this data. Furthermore, compound words in German are handled better in this way, since we have a more direct mapping between them and English words. We only present results with translations obtained with the combination of orthographic similarity and bigrams. Adding bigrams, improved upon the orthographic similarity translations by 0.92 for de-en and 0.75 for en-de. Using this technique, we obtain the highest performance on both translations directions.

We also extracted pseudo parallel sentences by mining NewsCrawl 2015. The similarity of a sentence pair is computed by calculating the average similarity between all source-target pairwise word similarities. The similarity between a source and target word is computed based on the BWEs and the orthographic similarity. We extracted ≈8K sentences. We oversampled the dataset to the size of the monolingual data and used it at the beginning of the training. We also attempted to use the original 8K sentences as a last fine-tuning step. Both approaches did not provide for improvements over our best scoring system.

## 6 Analysis

In Table 3 we present examples and we compare German→English translations with the different contrastive setups we outline in the experimental results. We show the phenomena that we observed and discuss some of the challenges that the systems are still not able to overcome. This can be a useful analysis that can provide insight into where future work should focus on.

In the first example we see that the models are to some extent able to do simple reorderings and insertions. We can see that most models were able to properly reorder "wollte die 45-Jährige" to "the 45-year-old wanted". The *Orth. + bigram* and *NewsCrawl 2017* were able to move "beruhigen" (calm) in front of "their brother" and furthermore inserted the preposition "to".

In the second example, we can observe that the models were again able to infer that the phrase "tot aufgefunden" should be reordered to "found dead". Additionally, the whole phrase was inserted at a much more appropriate place in the English sentence rather than at the end. Another interesting phenomenon is that the *NewsCrawl 2017* model was able to do a 2-1 mapping by translating "Einkaufzentrums" to "shopping centre". On the other hand, this example shows the challenges our models encounter. Given the relatively unintuitive mapping between "Koch" and "Hopkinson" that we have from the BWEs, the models had difficulty properly translating this word. Furthermore, most of them were not able to infer that "nach" in combination with "gezogen" translates to "moved to" and we see some more literal translations.

The third example shows some of the issues we had with translating named entities. Models without the orthographic similarity extension had trouble finding a suitable translation of "Erdogans". Furthermore most of the models inferred that adding the preposition "of" is necessary in this case.

The last example shows the importance of the dataset being used. The first three systems are trained on the same data and didn't translate "Kalendar" as opposed to the one trained on a subsample of NewsCrawl 2017. Although not necessarily related to the dataset being more recent, it shows that it most likely contained sentences that enabled proper translation to "calendar".

| | |
|---|---|
| *source* | Gemeinsam mit ihrem Lebensgefährten wollte die 45-Jährige ihren Bruder *beruhigen*. |
| *reference* | The 45-year-old and her partner wanted *to calm* down her brother. |
| *Final submission* | Met with her boyfriend, the 45-year-old wanted their brother calming. |
| *Orthographic* | Watching her boyfriend, the 45-year-old didn't have handled their brother. |
| *Orth. + bigram* | Together with her boyfriend, the 45-year-old wanted *to calm* their brother. |
| *NewsCrawl 2017* | Together with her boyfriend, the 45-year-old wanted *to calming* their brother. |
| *source* | Ein 28-jähriger Koch, der vor kurzem nach San Francisco gezogen ist, wurde im Treppenhaus eines örtlichen *Einkaufzentrums* tot aufgefunden. |
| *reference* | A 28-year-old chef who had recently moved to San Francisco was found dead in the stairwell of a local mall this week. |
| *Final submission* | A 28-yard Koch, who was pulled before he was pulled after San Francisco, was found in the stairwell of a local outlet dead Province. |
| *Orthographic* | A 28-year-old Reid, who has ever been relocated after San Francisco, was found dead in the hallway of a local crop. |
| *Orth. + bigram* | A 28-yard Koch, who recently moved after San Francisco, was found dead in the hallway of its local outlet. |
| *NewsCrawl 2017* | A 28-year-old Koch, who was given her home to San Francisco, was found dead in the stairwell of a local *shopping centre*. |
| *source* | Der Sport ist - wie das ganze Land - gespalten in Anhänger und Gegner Erdogans. |
| *reference* | The sport - like the entire country - is divided into those who support Erdogan, and those who do not. |
| *Final submission* | The sport is - like the whole country - divided in supporters and opponents Drogba. |
| *Orthographic* | The BBC is - like the whole country - divided in supporters and opponents of Erdogan. |
| *Orth. + bigram* | The sports is - like the whole country - divided in supporters and opponents of Erdogan. |
| *NewsCrawl 2017* | The sport is - like the whole country - divided in supporters and opponents of Mrs. May. |
| *source* | Das Treats Magazin arbeitet mit dem Fotografen David Bellemere zusammen, um einen 1970er Jahre Pirelli-inspirierten Kalendar für 2017 herauszubringen. |
| *reference* | Treats magazine is partnering with photographer David Bellemere to launch a 1970s' Pirelli-inspired calendar for 2017. |
| *Final submission* | The Treats magazine works with the photographers David Bellemere together, when a 1970s Pirelli-inspiring Kalendar for 2017 dates. |
| *Orthographic* | The Treats magazine works with the photographers David Bellemere together to bring a 1970s Pirelli-inspiring Kalendar for 2017. |
| *Orth. + bigram* | The Treats magazine works with the photographers David Bellemere together to bring a 1970s Pirelli-inspected Kalendar for 2017. |
| *NewsCrawl 2017* | The Treats magazine works with the brains David Bellemere together to attribute a 1970s Pirelli inspires calendar for 2017. |

Table 3: Example translations obtained using the different neural systems.

## 7 Conclusion

Corpus-based machine translation approaches typically require parallel training data. In this work, we have investigated methods which allow for unsupervised learning of translation models, i.e., we have examined how machine translation systems can be trained without any parallel data.

LMU Munich is one of two teams who participated in the WMT18 unsupervised learning subtrack for machine translation of news articles between German and English. Our shared task submission consists of an unsupervised phrase-based translation system and an unsupervised neural machine translation system.

We have shown how bigrams and orthographic similarity in the underlying bilingual word embeddings benefit the results. We have presented effective unsupervised learning techniques for both the phrase-based and the neural paradigm and have demonstrated how an effective training curriculum improves translation quality.

## Acknowledgments

We thank Helmut Schmid for helpful discussions and comments. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement № 640550)

## References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised Neural Machine Translation. In *International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*, ICLR '15. ArXiv: 1409.0473.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46.

Fabienne Braune, Viktor Hangya, Tobias Eder, and Alexander Fraser. 2018. Evaluating Bilingual Word Embeddings on the Long Tail. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 188–193.

Fabienne Cap, Marion Weller, Anita Ramm, and Alexander Fraser. 2014. CimS – The CIS and IMS Joint Submission to WMT 2014 Translating from English into German . In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 71–78.

Fabienne Cap, Marion Weller, Anita Ramm, and Alexander Fraser. 2015. CimS - The CIS and IMS Joint Submission to WMT 2015 Addressing Morphological and Syntactic Differences in English to German SMT . In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 84–91.

Colin Cherry and George Foster. 2012. 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, Montréal, Canada. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word Translation Without Parallel Data. *CoRR*, abs/1710.04087.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving Human Parity on Automatic Chinese to English News Translation. *arXiv preprint arXiv:1803.05567*.

Matthias Huck, Fabienne Braune, and Alexander Fraser. 2017. LMU Munich's Neural Machine Translation Systems for News Articles and Health Information Texts. In *Proceedings of the Second Conference on Machine Translation*, pages 315–322.

Matthias Huck, Alexander Fraser, and Barry Haddow. 2016. The Edinburgh/LMU Hierarchical Machine Translation System for WMT 2016. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 311–318.

Matthias Huck, Dario Stojanovski, Viktor Hangya, and Alexander Fraser. 2018. LMU Munich's Neural Machine Translation Systems at WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Volume 2, Shared Task Papers*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Philipp Koehn and Rebecca Knowles. 2017. Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised Machine Translation Using Monolingual Corpora Only. In *International Conference on Learning Representations*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-Based & Neural Unsupervised Machine Translation. *arXiv preprint arXiv:1804.07755*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, USA. Association for Computational Linguistics.

Jan-Thorsten Peter, Tamer Alkhouli, Hermann Ney, Matthias Huck, Fabienne Braune, Alexander Fraser, Aleš Tamchyna, Ondřej Bojar, Barry Haddow, Rico Sennrich, Frédéric Blain, Lucia Specia, Jan Niehues, Alex Waibel, Alexandre Allauzen, Lauriane Aufrant, Franck Burlot, elena knyazeva, Thomas Lavergne, François Yvon, Mārcis Pinnis, and Stella Frank. 2016. The QT21/HimL Combined Machine Translation System. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 344–355.

Hassan Sajjad, Svetlana Smekalova, Nadir Durrani, Alexander Fraser, and Helmut Schmid. 2013. QCRI-MES Submission at WMT13: Using Transliteration Mining to Improve Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 219–224.

Aleš Tamchyna, Roman Sudarikov, Ondřej Bojar, and Alexander Fraser. 2016. CUNI-LMU Submissions in WMT2016: Chimera Constrained and Beaten. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 385–390.

Marion Weller, Max Kisselew, Svetlana Smekalova, Alexander Fraser, Helmut Schmid, Nadir Durrani, Hassan Sajjad, and Richárd Farkas. 2013. Munich-Edinburgh-Stuttgart Submissions at WMT13: Morphological and Syntactic Processing for SMT. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 232–239.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation . In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011.

# Tencent Neural Machine Translation Systems for WMT18

**Mingxuan Wang, Li Gong, Wenhuan Zhu, Jun Xie, Chao Bian**
Mobile Internet Group, Tencent Technology Co., Ltd
{xuanswang,ligong,wenhuanzhu,stiffxie,chaobian}@tencent.com

## Abstract

We participated in the WMT 2018 shared news translation task on English↔Chinese language pair. Our systems are based on attentional sequence-to-sequence models with some form of recursion and self-attention. Some data augmentation methods are also introduced to improve the translation performance. The best translation result is obtained with ensemble and reranking techniques. Our Chinese→English system achieved the highest cased BLEU score among all 16 submitted systems, and our English→Chinese system ranked the third out of 18 submitted systems.

## 1 Introduction

In recent years, the emergence of seq2seq models has revolutionized the field of MT by replacing traditional phrase-based approaches with neural machine translation (NMT) systems based on the encoder-decoder paradigm. A successful extension of encoder-decoder models is the attention mechanism which conducts a soft search over source tokens and yields an attentive vector to represent the most relevant segments of the source sentence for the current decoding state (Luong et al., 2015; Bahdanau et al., 2014; Wu et al., 2016; Sutskever et al., 2014; Tu et al., 2016; Zhou et al., 2016). Most recently, the Transformer model, which is based solely on a self-attention mechanism and feed-forward connections, has further advanced the field of NMT, both in terms of translation quality and speed of convergence(Vaswani et al., 2017; Ahmed et al., 2018). In this paper, we describe the Tencent NMT (TNMT) systems submissions for the WMT 2018 Chinese→English and English→Chinese translation task.

We propose two different architectures as our end to end approaches namely RNMT and Transformer. For RNMT, we implemented a hybrid multi-layer attention-based encoder-decoder model. The decoder was implemented as Recurrent Neural Networks (RNNs) and the encoder was represented with self-attention layers. We also integrated with some recent promising techniques in RNMT including the methods which made significantly contribution to the success of Transformer. In doing so, we come up with an enhanced version of RNMT that achieves comparable performance with Transformer. For Transformer, we follow the latest version of the Transformer model in the public Tensor2Tensor[1] codebase. The Transformer model replaces the recurrent connections with self-attention which can be taken as a complement with the RNMT model.

For data augmentation, we used automatic back-translation of a sub-selected monolingual News corpus as additional training data(Sennrich et al., 2015). To achieve strong machine translation performance, we further leverage the joint training method described in (Hassan et al., 2018) to optimize both the target-to-source (T2S) and source-to-target (S2T) model by extending

[1] https://github.com/tensorflow/tensor2tensor

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 522–527
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64056

the back-translation method. The joint training method uses both the monolingual and bilingual data and updates NMT models through several iterations. We also apply several knowledge distillation methods to leverage the information gain of different architectures. To alleviate the exposure bias problem of the *left-to-right* (L2R) model, Agreement Regularization was introduced as a teacher network (Hassan et al., 2018; Liu et al., 2016). Ensemble teacher networks and architecture teacher networks are also introduced to boost the performance of a single model.

In addition, we consider the system combination and improve the performance by reranking (Koehn et al., 2003) the $n$-best translation outputs of the ensemble models with some effective features, including the *target-to-source* (T2S) score, *left-to-right* (L2R) score, *right-to-left* (R2L) score, Transformer score and RNMT score. The ensemble models are trained with different architectures or parameter settings to increase the diversity of the system. As a result, our Chinese→English system achieved the highest cased BLEU score among all 16 submitted systems, and our English→Chinese system ranked the third out of 18 submitted systems.

## 2 NMT Baseline System

We apply two different NMT architectures for the shared news translation task as our baseline systems.

1. RNMT: A hybrid deep attentional encoder-decoder networks with a stack Long Short Term Memory (LSTM) recurrent neural network for decoder and a deep self-attention network for encoder. Inspired by Transformer, Multi-head additive attention is used instead of the single-head attention in the RNMT model. Layer normalization is also applied within the output of LSTM cells. In our setup, the dimension of word embeddings and the hidden layers are both set to 1024. The encoder has 6 self-attention layers and the decoder has 3 LSTM layers.
2. Transformer: Our reimplementation of tensor2tensor with minor changes. We also implement a C++ version of the system for speeding up the decoding process. The default parameters of Transformer Big model is adopted as our transformer baseline and we further change the hyper-parameters to find the best settings on the develop set.

We train the models with adadelta(Zeiler, 2012), reshuffling the training corpus between epochs. We batched sentence pairs by approximate length, and limited input and output tokens per batch to 8192 per GPU. Each resulting training batch contained approximately $60,000$ source and $60,000$ target tokens. To avoid gradient explosion, the gradients of the cost function which had $\ell_2$ norm larger than a predefined threshold $25$ were normalized to the threshold. During training, we employed label smoothing of value ranging from $0.05$ to $0.2$ and set dropout rate from $0.01$ to $0.3$(Hinton et al., 2012; Pereyra et al., 2017). We perform early stopping on the baseline system and validate the model every $1000$ mini-batches against BLEU on the WMT 17 news translation test set.

## 3 Experiment Techniques

### 3.1 Back Translation

In statistical machine translation, large monolingual corpora in the output language have traditionally been used for training language models to make the system output more fluent. However, it is difficult to integrate language models in current NMT architectures. Instead of ignoring such large monolingual corpora, Sennrich et al. (2015) exploited large corpora in the output language by translating a subset of them into the input language and then using the resulting synthetic sentence pairs as additional training data. We translated monolingual English text into Chinese using our English→Chinese system and translated monolingual Chinese text into English using our Chinese→English system described in Section 2. To improve the quality of the synthetic corpus we propose to use the ensemble models to translate the target sentence.

To select sentences for back-translation, we used semi-supervised convolutional neural network classifiers (Chen et al., 2017) and LSTM language models respectively. We selected 80M sentences from the target monolingual corpus based on both their classifier and language model scores, which reflect their similarity to the in-domain corpus. The selected sentences are then translated and divided into 8 portions with each contains 10M synthetic sentence pairs. Each portion is used to enhance an individual baseline model.

### 3.2 Joint Training of Source-to-Target and Target-to-Source Models

Back translation augments parallel data with plentiful monolingual data, allowing us to train source-to-target (S2T) models with the help of target-to-source (T2S) models. In order to leverage both source and target language monolingual data, and also let S2T and T2S models help each other, we leverage the joint training method to optimize them by extending the back-translation method(Zhang et al., 2018).

The joint training method uses both the source and the target monolingual data and updates NMT models through several iterations. In iteration 1, the process can be viewed as traditional back translation methods. The T2S model translated the target monolingual data to help the S2T model. Similarly, we can optimize the T2S translation model with the help of S2T translation model. In iteration 2, the above process is repeated, and the synthetic training data are re-translated with the updated T2S and S2T model. It is worth noting that ensemble models are used to generate the synthetic corpus so that the negative impact of noisy translations can be minimized. In order to increase the robustness of the system, we also re-translated the target of the bilingual corpus as the synthetic data. The joint training process continues until the performance on a development data set is no longer improved. We repeated three iterations for all our systems.

### 3.3 Knowledge Distillation

Knowledge distillation describes a method for training a student network to perform better by learning from a stronger teacher network. In our experiments, it is surprising to find that the teacher network is not necessarily stronger than the student network. The student network is capable of learning complementary information from even a worser heterogeneous teacher. We therefore investigated three different kinds of teacher networks to enhance the translation performance of a student NMT network.

**R2L Teacher** The approach is also referred as *Agreement Regularization of Left-to-Right and Right-to-Left Models* to integrate the information of R2L models to L2R ones (Hassan et al., 2018) . Following this work, we translate the source sentences of the parallel data with R2L model and use the translated pseudo corpus to improve the L2R model. It is worth noting that we filter the pseudo corpus with BLEU score lower than 30.

**Ensemble Teacher** We also apply knowledge distillation on ensemble teacher models (Freitag et al., 2017). Similar with R2L teacher model, we use ensemble models to translate the source side sentence of the parallel corpus and then apply the pseudo corpus to the training corpus.

**Architecture Teacher** The RNMT and Transformer models achieve similar performances but use very different ways to encode and decode context which leverage the advantages by combine the information of both architectures. We therefore use a teacher network to boost a student network with different arctectures.

### 3.4 System Combination and Re-ranking

For single models, we average the last 60 checkpoints to avoid overfitting. The checkpoint are saved every 600 seconds. For ensemble models, we trained 8 systems with different parameters and the different portion of monolingual corpus selected in Section 3.1. Since both the source and target sentences can be generated from left to right and from right to left, we can have a total of eight ensemble systems, which including RNMT-S2T-R2L, RNMT-S2T-L2R, Transformer-S2T-L2R, Transformer-S2T-R2L, RNMT-T2S-R2L, RNMT-T2S-L2R, Transformer-T2S-L2R and Transformer-T2S-R2L.

For both S2T and T2S direction, we rescored 200-best lists output from four ensemble systems (S2T or T2S) using a rescoring model consisting of eight features: four S2T ensemble model scores and four T2S ensemble model scores.

## 4 Experiments Settings and Results

### 4.1 Pre-processing and Post-processing

We first segmented the Chinese sentences with our Chinese word segmentation tool and tokenized English sentences with the scripts provided in Mosess[2]. To enable open-vocabulary, we use BPE (Sennrich et al., 2016) with 50K operations. In our preliminary experiments, we found that BPE works better than UNK replacement techniques. We also filter bad sentences according to the alignment score obtained by fast-align toolkit [3] and remove duplications in the training data. The preprocessed training data consists of 19M bilingual pairs.

For Chinese→English translation, the final output was true-cased and de-tokenized with the scripts provided in Moses. For English→Chinese translation, we normalized the punctuations of the outputs with our in-house script and remove the space between the Chinese characters.

### 4.2 Chinese→English Systems

Table 1 shows the Chinese→English translation results on validation set (WMT2017). We reported cased BLEU scores calculated with Moses mteval-v13a.pl script[4]. The Transformer and RNMT model achieved similar results in terms of the mean BLEU scores which is consistent with the observations of Chen et al., (2018). In order to obtain more diverse models and better ensemble results, we trained eight models independently with different random initializations and dropout rate ranging from 0.01 to 0.3.

| SYSTEM | BLEU |
|---|---|
| RNMT | |
| Baseline | 24.2 |
| + Back Translation | 25.4 |
| + Joint Training | 26.1 |
| + R2L Teacher | 27.1 |
| + Transformer Teacher | 27.3 |
| + Ensemble Teacher | 27.7 |
| Transformer | |
| Baseline | 24.3 |
| +ALL features | 27.6 |
| System Combination | |
| Ensemble Baseline + Rerank | 26.1 |
| Ensemble BT + Rerank | 27.2 |
| Ensemble Best | 27.9 |
| Ensemble Best + Rerank | 28.5 |

**Table 1:** Chinese→English Systems BLEU results on development set (WMT17). Submitted system is the last system.

The synthetic data plays an import role in the success of our system. As for the single model, back translation improved the strong baseline by 1.2 BLEU score. Even for system combination, the synthetic data still achieved a stable improvements from 26.1 to 27.2 in terms of BLEU. As an extension of the back translation method, the joint training approach interactively makes data augmentation by boosting source-to-target and target-to-source NMT systems. The method again obtained a substantial improvements up to 0.7 BLEU score.

Among knowledge distillation methods, the R2L teacher significantly enhanced our single system by 1.0 BLEU score. The Transformer teacher and ensemble teacher further get an improvements by 0.2 and 0.4 in terms of BLEU.

Applying different combinations of the techniques described in Section 3.4, we build eight single systems with all the optimization techniques described in Section 3. We then obtained

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

[3] https://github.com/clab/fast_align

[4] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v13a.pl

four ensemble models including Transformer-L2R, Transformer-R2L, RNMT-L2R and RNMT-R2L. We then rescored 800 best lists output from the our ensemble NMT systems using a rescoring mode consisting of eight features. As can be seen in the Table 1. After ensemble a little improvement over the best single model by 0.2 BLEU is achieved. One possible explanation is that the information gain of the ensemble model has been obtained by the distillation method. For rerank model, we finally achieved an improvements of 0.6 BLEU score with fine-tuned feature weights.

### 4.3 English→Chinese Systems

| SYSTEM | BLEU |
|---|---|
| RNMT | |
| Baseline | 35.9 |
| + Joint Training | 38.5 |
| + ALL features | 40.1 |
| Transformer | |
| Baseline | 35.0 |
| +ALL features | 39.8 |
| System Combination | |
| Ensemble Best | 40.4 |
| Ensemble Best + Rerank | 41.1 |

**Table 2:** English→Chinese Systems BLEU results on development set (WMT17). Submitted system is the last system.

Table 2 shows the English→Chinese translation results on development set. All results are evaluated by character-level BLEU. We followed exactly the same settings with the Chinese→English translation system. In this case, the Joint Training method brought a substantial improvement over 2.6 BLEU scores showing the advantages of using the monolingual data and integrating the S2T model and T2S model. For knowledge distillation, We observed an improvement of 1.6 BLEU score. Finally, we applied ensemble and reranking methods, which provided 1.3 BLEU improvements over the best single model.

## 5 Conclusion

We present the *Tencent* NMT systems for WMT 2018 Chinese↔English news translation tasks. For both translation directions, our final systems achieved substantial improvements up by $4 \sim 5$ BLEU score over baseline systems by integrating the following technique:

1. Back translation the target monolingual data set
2. Joint training of the S2T and T2S systems
3. Knowledge distillation with R2L teacher networks, architecture teacher networks and ensemble teacher networks
4. System combination and reranking.

As a result, our submitted Chinese→English system achieved the highest cased BLEU score among all 16 submitted systems and our English→Chinese system ranked the third out of 18 submitted systems.

## References

[Ahmed et al.2018] Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. 2018. Weighted transformer network for machine translation. *arXiv: Artificial Intelligence*.

[Bahdanau et al.2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Chen et al.2017] Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 40–46.

[Chen et al.2018] Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George F Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. *meeting of the association for computational linguistics*.

[Freitag et al.2017] Markus Freitag, Yaser Alonaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv: Computation and Language*.

[Hassan et al.2018] Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

[Hinton et al.2012] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors.

[Koehn et al.2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. 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.

[Liu et al.2016] Lemao Liu, Masao Utiyama, Andrew M Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. pages 411–416.

[Luong et al.2015] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

[Pereyra et al.2017] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv: Neural and Evolutionary Computing*.

[Sennrich et al.2015] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

[Sennrich et al.2016] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. *meeting of the association for computational linguistics*, 1:1715–1725.

[Sutskever et al.2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[Tu et al.2016] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. *ArXiv eprints, January*.

[Vaswani et al.2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Llion Jones, Jakob Uszkoreit, Aidan N Gomez, and ukasz Kaiser. 2017. Attention is all you need. *neural information processing systems*, pages 5998–6008.

[Wu et al.2016] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[Zeiler2012] Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

[Zhang et al.2018] Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. *national conference on artificial intelligence*.

[Zhou et al.2016] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *arXiv preprint arXiv:1606.04199*.

# The NiuTrans Machine Translation System for WMT18

**Qiang Wang**[12], **Bei Li**[1], **Jiqiang Liu**[1], **Bojian Jiang**[1],
**Zheyang Zhang**[1], **Yinqiao Li**[12], **Ye Lin**[1], **Tong Xiao**[12], **Jingbo Zhu**[12]
[1]Natural Language Processing Lab., Northeastern University
[2]NiuTrans Co., Ltd., Shenyang, China
wangqiangneu@gmail.com, {xiaotong, zhujingbo}@mail.neu.edu.cn
{libeinlp, liujiqiang, jiangbojian}@stumail.neu.edu.cn
{zhangzheyang, liyinqiao, linyeneu}@stumail.neu.edu.cn

## Abstract

This paper describes the submission of the *NiuTrans* neural machine translation system for the WMT 2018 Chinese ↔ English news translation tasks. Our baseline systems are based on the Transformer architecture. We further improve the translation performance 2.4-2.6 BLEU points from four aspects, including architectural improvements, diverse ensemble decoding, reranking, and post-processing. Among constrained submissions, we rank 2nd out of 16 submitted systems on Chinese → English task and 3rd out of 16 on English → Chinese task, respectively.

## 1 Introduction

Neural machine translation (NMT) exploits an encoder-decoder framework to model the whole translation process in an end-to-end fashion, and has achieved state-of-the-art performance in many language pairs (Wu et al., 2016; Sennrich et al., 2016c). This paper describes the submission of the *NiuTrans* neural machine translation system for the WMT 2018 Chinese ↔ English news translation tasks.

Our baseline systems are based on the Transformer model due to the excellent translation performance and fast training thanks to the self-attention mechanism. Then we enhance it with checkpoint ensemble (Sennrich et al., 2016c) that averages the last N checkpoints of a single training run. To enable open-vocabulary translation, all the words are segmented via byte pair encoding (BPE) (Sennrich et al., 2016b) for both Chinese and English. Also, we use back-translation technique (Sennrich et al., 2016a) to leverage the rich monolingual resource.

Beyond the baseline, we achieve further improvement from four aspects, including architectural improvements, diverse ensemble decoding, reranking and post-processing. For architectural improvements, we add relu dropout and attention dropout to improve the generalization ability and increase the inner dimension of feed-forward neural network to enlarge the model capacity (Hassan et al., 2018). We also use the novel Swish activation function (Ramachandran et al., 2018) and self-attention with relative positional representations (Shaw et al., 2018). Next, we explore more diverse ensemble decoding via increasing the number of models and using the models generated by different ways. Furthermore, at most 17 features tuned by MIRA (Chiang et al., 2008) are used to rerank the N-best hypotheses. At last, a post-processing algorithmic is proposed to correct the inconsistent English literals between the source and target sentence.

Through these techniques, we can achieve 2.4-2.6 BLEU points improvement over the baselines. As a result, our systems rank the second out of 16 submitted systems on Chinese → English task and the third out of 16 on English → Chinese task among constrained submissions, respectively.

## 2 Baseline System

Our systems are based on Transformer (Vaswani et al., 2017) implemented on the Tensor2Tensor [1]. We use base Transformer model as described in (Vaswani et al., 2017): 6 blocks in the encoder and decoder networks respectively (word representations of size 512,

[1] https://github.com/tensorflow/tensor2tensor/tree/v1.0.14. We choose this version because we found that this implementation is more similar to the original model described in (Vaswani et al., 2017) than newer versions.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 528–534
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64057

feed-forward layers with inner dimension 2048, 8 attention heads, residual dropout is set to 0.1). We use negative Maximum Likelihood Estimation (MLE) as loss function, and train all the models using Adam with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. The learning rate is scheduled as described in (Vaswani et al., 2017): $lr = d^{-0.5} \cdot min(t^{-0.5}, t \cdot 4000^{-1.5})$, where $d$ is the dimension of word embedding, $t$ is the training step number. To enable the open-vocabulary translation, we use byte pair encoding (BPE) (Sennrich et al., 2016b) for both Chinese and English. All the models are trained for 15 epochs on one machine with 8 NVIDIA 1080 Ti GPUs. We limit source and target tokens per batch to 4096 per GPU, resulting in approximate 25,000 source and 25,000 target tokens in one training batch. We also use checkpoint ensemble by averaging the last 15 checkpoints, which are saved at 10-minute intervals.

For evaluation, we use beam search with length normalization (Wu et al., 2016). By default, we use beam size of 12, while the coefficient of length normalization is tuned on development set. We use the home-made C++ decoder as a more efficient alternative to the tensorflow implementation, which is also necessary for our diverse ensemble decoding (Section 3.2). The hypotheses that own too many consecutive repeated tokens (e.g. beyond the count of the most frequent token in the source sentence) are removed. We report all experimental results on *newsdev2018* by the official evaluation tool *mteval-v13a.pl*.

## 3 Improvements

We improve the baseline system from four aspects, including architectural improvements, ensemble decoding, reranking and post-processing.

### 3.1 Architectural Improvements

**Dropout** The original Transformer only uses residual dropout when the information flow is added between two adjacent layers/sublayers, while the dropouts in feed-forward neural network (e.g. relu dropout) and self attention weights (e.g. attention dropout) are not in use. In practice, we observed the consistent improvements than baseline when we set relu dropout to 0.1 and attention dropout to 0.1, thanks to the regularization effect to overcome the overfitting.

**Larger Feed-Forward Network** Limited by the size of GPU memory, we can not directly train a big Transformer model with the batch size as large as the base model. To solve this, we resort to increase the inner dimension (refer to $d_{ff}$) of feed-forward network while other settings stay the same. It is consistent with the finding of (Hassan et al., 2018) that the transformer model can benefit from larger $d_{ff}$.

**Swish Activation Function** The standard Transformer model has a non-linear expression capability due to the use of Rectified Linear Unit (ReLU) activation function. Recently, Ramachandran et al. (2018) propose a new activation function called Swish by the network automatic search techniques based on reinforcement-learning. They claim that Swish tends to work better than ReLU on deeper models and can transfer well to a number of challenging tasks. Formally, Swish is computed as:

$$Swish(x) = x \cdot sigmoid(\beta x),$$

where $\beta$ is either a constant or a learnable parameter. In practice, we replace ReLU with Swish ($\beta = 1$) and do not change any other settings.

**Relative Positional Representation** Transformer uses the absolute position encodings based on sinusoids of varying frequency, while Shaw et al. (2018) point out that the representations of relative position can yield consistent improvement over the absolute counterpart. They equip the representations of both key and value with some trainable parameters (e.g. $a_{ij}^K$, $a_{ij}^V$ in (Shaw et al., 2018)) when calculating the self attention. We re-implement this model, and use clipping distance $k = 16$ with the unique edge representations per layer and head. We use both the absolute and relative positional representations simultaneously.

### 3.2 Diverse Ensemble Decoding

Ensemble decoding is a widely used technique to boost the performance by integrating the predictions of several models, and has been

| | |
|---|---|
| **Source:** | 于是就有了这个去年 9 月发布的P@@ ass@@ p@@ ort 。 |
| **Translation:** | so there is the Pas@@ port , which was released last September . |
| **Post-Processing:** | so there is the Passport , which was released last September . |
| **Source:** | Furious residents have savaged Sol@@ i@@ hull Council saying it was “ useless at dealing with the problem ”. |
| **Translation:** | 愤怒的居民猛烈抨击了 S@@ ol@@ i@@ h@@ ou@@ s@@ 委员会, 称它 “ 在处理这个问题上是无用的” 。 |
| **Post-Processing:** | 愤怒的居民猛烈抨击了 Solihull 委员会, 称它 “ 在处理这个问题上是无用的” 。 |

Table 1: Samples of the inconsistent translation of the constant literal between source and target sentence. The subword is split by “@@”. The two samples are picked up from *newstest2018*.

proved effective in the WMT competitions (Sennrich and Haddow, 2016; Sennrich et al., 2017; Wang et al., 2017). Existing experimental results about ensemble decoding mainly concentrate upon a small number of models (e.g. 4 models (Wang et al., 2017; Sennrich et al., 2016c, 2017)). Besides, the ensembled models generally lack of sufficient diversity, for example, Sennrich et al. (2016c) use the last N checkpoints of a single training run, while Wang et al. (2017) use the same network architecture with different random initializations.

In this paper, we study the effects of more diverse ensemble decoding from two perspectives: the number of models and the diversity of integrated models. We explore at most 15 models for jointly decoding by allocating two models per GPU device in our C++ decoder. In addition to using different random seeds, the ensembled models are generated from more diverse ways, such as different training steps, model sizes and network architectures (see Section 3.1).

Every ensembled model is also assigned a weight to indicate the confidence of prediction. In practice, we simply assign the same weight 1.0 for each model. We also study the greedy tuning strategy (randomly initialize all weights firstly, then fix other weights and only tune one weight each time), while there is no significant improvement observed. [2]

### 3.3 Reranking

We apply the reranking module to pick up a potentially better hypothesis from the n-best generated by ensemble decoding. The used features for reranking include:

- TFs: Translation features. We totally use eight types of translation features, and each type can be represented as a tuple with four elements: ($L_s$, $D_s$, $L_t$, $D_t$), where $L_s, L_t \in \{ZH, EN\}$ denotes the language of source and target respectively, and $D_s, D_t \in \{L2R, R2L\}$ denotes the direction of source and target sequence respectively. For example, (ZH, L2R, EN, R2L) denotes a system trained on ordinal Chinese → reversed English.

- LM: 5-gram language model of target side [3].

- SM: Sentence similarity. The best hypothesis from the target R2L system is compared to each n-best hypothesis and used to generate a sentence similarity score based on the cosine of the two sentence vectors. The sentence vector is represented by the mean of all word embeddings.

Given the above features, we calculate the ranking score by a simple linear model. All weights are tuned on the development set via MIRA. The hypothesis with the highest ranking score is chosen as the refined translation.

### 3.4 Post-Processing

Current NMT system generates the translation word by word [4], which is difficult to guarantee the consistency of some constant literals between source sentence and its translation.

In this section, we focus on the English literals in a Chinese sentence. For example, as

[2] We do not use some more sophisticated tuning methods, such as MERT, MIRA, due to the expensive cost for ensemble decoding, especially with a large beam size.

[3] All language models are trained by KenLM (Heafield, 2011).

[4] Actually it is subword by subword in this paper.

**Algorithm 1** Post-processing algorithmic for inconsistent English literals translation.

```
Input: $S$: source sentence; $T$: NMT translation;
Output: $T'$: translation after post-processing
Initialize: $T' = T$, create $\mathbb{S}(x, y)$ saves the similarity between $x$ and $y$
Get the set of English literals $\mathbb{EL}$ from Chinese sentence (either $S$ or $T$)
for each English literal $el$ in $\mathbb{EL}$ do
    if $el$ not in $T$ then
        for each $y$ in the set of $n$-gram of $T$ $(1 \leq n \leq 3)$ do
            $\mathbb{S}(el, y) = sim(el, y)$
        end for
    end if
    $y^* = argmax_y \mathbb{S}(el, y)$
    replace $el$ with $y^*$ in $T'$
end for
```

shown in Table 3.2, the literal “Passport” in Chinese sentence is translated into “Pasport” wrongly, and a similar error happens between “Solihull” and its translation “Solihous”.

To solve this issue, we propose a post-processing method to correct the unmatched translations for the constant literals, as shown in Algorithm 1. The basic idea is that the English literals appearing in Chinese sentence must be contained in English sentence. The challenge is that how to align the correct literal with its wrong one. In practice, we compute the normalized edit distance as the similarity:

$$sim(x, y) = \frac{D(x, y)}{L_x}, \tag{1}$$

where $D(x, y)$ denotes the edit-distance between $x$ and $y$, $L_x$ is the length of $x$. Then, the most similar translated literal is recovered by the original one.

Since the number of Chinese sentences containing the English literals is relatively small, our approach can not significantly improve the BLEU, but we find that it is very effective for human evaluation.

## 4 Experiments and Results

### 4.1 Chinese → English Results

For Chinese → English task, we use all the CWMT corpus and partial of UN and News-Commentary combined corpus [5]. We also augment the training data by back-translation of the *NewsCraw2017* corpus using the baseline system based on the parallel data only. All texts are segmented by home-made word segmentation toolkit [6]. We remove the parallel sentence pairs which is duplicated, exceptional length ratio, or bad alignment score obtained by fast-align [7]. As a result, we use 7.2M CWMT corpus, 4.2M UN and News-Commentary combined corpus, and 5M pseudo parallel data. Detailed statistical information of training data is shown in Table 2. Then we learn BPE codes with 32k merge operations from independent Chinese and English text, resulting in the size of source and target vocabulary is 47K and 33K respectively. We also study the effect of merge operations, however no significant gain is found when we shrink or expand the number of merge operations.

Table 3 presents the BLEU scores on *newsdev2018* for Chinese → English task. Firstly, we can see that using checkpoint ensemble brings +0.82 BLEU than the baseline of single model. When we equip the Transformer base model with larger $d_{ff}$ and relu & attention dropout, +0.56 BLEU are improved further. However, to our disappointment, we do not observe consistent improvement via Swish or relative positional representations.

Based on the strong single model baseline, we firstly study the conventional ensemble decoding: 4 models with different random seeds, resulting in a significant gain of 0.72 BLEU point. Then we use 4 models with different architectures: *baseline*, $d_{ff}$ = 4096, *dropout* and $d_{ff}$=*4096 + dropout*, then an interesting result is that the diverse ensemble decoding is superior than the ensemble of $d_{ff}$ + *dropout*, which provides an evidence that diverse models may be more important than homogeneous strong models. The beam size of 100 is a bit better than 12. This result is inconsistent with previous work claiming that larger beam size can badly drop down the performance (Tu

[5] We randomly sample 30% data, and found that it can achieve comparable performance with the full data. In this way, we can train more models for our diverse ensemble decoding and reranking.

[6] For Chinese, the word segmentation is done based on unigram language model with Viterbi algorithm.

[7] https://github.com/clab/fast_align

| Direction | Lang. | Sentences | Tokens | Ave. sentence length |
|---|---|---|---|---|
| ZH → EN | ZH | 16.5M | 391M | 23.7 |
| | EN | 16.5M | 415M | 25.2 |
| EN → ZH | EN | 16.9M | 505M | 29.9 |
| | ZH | 16.9M | 465M | 27.5 |

Table 2: Statistics of the training data

| System | | beam size | Valid. |
|---|---|---|---|
| Baselines | Transformer-Base | 12 | 25.09 |
| | +checkpoint ensemble | 12 | 25.91 |
| Architectural Improvements | $+d_{ff}$ =4096 | 12 | 26.17 |
| | +dropout | 12 | 26.45 |
| Diverse Decoding | 4 same models with different random seeds | 12 | 27.21 |
| | 4 diverse models | 12 | 27.67 |
| | 4 diverse models with large beam | 100 | 27.69 |
| | 8 diverse models | 100 | 28.06 |
| | 15 diverse models | 80 | 28.18 |
| Re-ranking | 14 features | - | 28.46 |
| Post-processing | **English literal revised*** | - | 28.46 |

Table 3: BLEU scores [%] on *newsdev2018* Chinese-English translation. * denotes the submitted system.

et al., 2017), which needs to be invested further. Additionally, we expand the number of models from 4 to 8 and 15 [8], the overall performances are further improved +0.35 and +0.52 respectively. For 15 models ensemble decoding, we arrange every two models on one GPU via our C++ decoder except the big model which requires one GPU.

Then we rerank the n-best from diverse ensemble decoding (at most 80 candidates) with 14 features [9], we achieve +0.28 BLEU improvement thanks to the complementary information brought by the features. At last, we do post-processing for the reranking output, but almost no effect on BLEU due to limited English literals are found in Chinese sentences.

### 4.2 English → Chinese Results

For English → Chinese translation, the training data also consists of three parts: CWMT corpus, part of UN and News-Commentary combined data and pseudo parallel data from back-translation. The differences from Chinese → English translation are that the UN and News-Commentary combined data is selected by XenC (Rousseau, 2013) [10] according to the *xmu* Chinese monolingual corpus from CWMT, and *xin_cmn* monolingual corpus is used for back-translation. Data preprocessing is same as Section 4.1, resulting in 7.2M CWMT corpus, 3.5M UN and News-Commentary combined corpus, and 6.2M pseudo parallel data. Then 32k merge operations are used for BPE.

Like Chinese → English, using checkpoint ensemble can bring a gain of +0.62 BLEU solidly. Besides, increasing the dimension of $d_{ff}$ and activate more dropout are proved effective again. The biggest difference from Chinese → English is that diverse ensemble decoding improves the performance at most +1.33 BLEU when we integrate 10 models. However, increasing either the number of models or the diversity is helpful for ensemble decoding. As for reranking, although we only use four (EN, ZH, L2R, R2L) models as features due to time constraint. there is still +0.35 BLEU improvement obtained. At last, post-processing makes an more obvious effect for English → Chinese translation than Chinese → English, because the BLEU4 is computed on characters rather than tokens.

[8] The types of used models include *baseline*, $d_{ff}$, *dropout*, $d_{ff}$ + *dropout*, *Swish*, *RPR* (relative position representation), *big* (Transformer big model with small batch size) and *baseline-epoch20* (training 20 epochs rather than 15).

[9] Four (ZH, EN, L2R, L2R) models, four (ZH, EN, L2R, R2L) models, one (ZH, EN, R2L, L2R) feature, one (ZH, EN, R2L, R2L) feature, one (EN, ZH, R2L, L2R) feature, one (EN,ZH,R2L,R2L) feature, one LM feature and one SM feature.

[10] https://github.com/antho-rousseau/XenC

| System | | beam size | Valid. |
|---|---|---|---|
| Baselines | Transformer-Base | 12 | 38.41 |
| | +checkpoint ensemble | 12 | 39.03 |
| Model Variance | +$d_{ff}$=4096 | 12 | 39.48 |
| | +dropout | 12 | 39.61 |
| Diverse Decoding | 4 same models with different random seeds | 12 | 40.19 |
| | 4 diverse models | 12 | 40.46 |
| | 4 diverse models + big beam | 50 | 40.54 |
| | 10 diverse models | 50 | 40.94 |
| Re-ranking | 4 features | - | 41.29 |
| Post-processing | **English literal revised*** | - | 41.41 |

Table 4: BLEU scores [%] on *newsdev2018* English → Chinese translation. * denotes the submitted system.

## 5 Conclusion

This paper presents the *NiuTrans* system to the WMT 2018 Chinese ↔ English news translation tasks. Our single model baseline use the Transformer architecture, and has achieve comparable performance than the last year's best ensembled results. We further improve the baseline's performance from four aspects, including architectural improvements, diverse ensemble decoding, reranking and post-processing. We find that increasing the number of models and the diversity of models is crucial for ensemble decoding. In addition, as the improvement of ensemble decoding, the gain from reranking gradually decreases. Among all the constrained submissions to the Chinese ↔ English news task, our submission is ranked 2nd out of 16 submitted systems on Chinese → English task and the 3rd out of 16 on English → Chinese task, respectively.

## Acknowledgments

This work was supported in part by the National Science Foundation of China (No. 61672138 and 61432013), the Fundamental Research Funds for the Central Universities.

## References

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In Proceedings of the conference on empirical methods in natural language processing, pages 224–233. Association for Computational Linguistics.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. arXiv preprint arXiv:1803.05567.

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.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2018. Searching for activation functions.

Anthony Rousseau. 2013. Xenc: An open-source tool for data selection in natural language processing. The Prague Bulletin of Mathematical Linguistics, (100):73–82.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh’ s neural mt systems for wmt17. WMT 2017, page 389.

Rico Sennrich and Barry Haddow. 2016. Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers, chapter Linguistic Input Features Improve Neural Machine Translation. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August

7-12, 2016, Berlin, Germany, Volume 1: Long Papers.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, chapter Edinburgh Neural Machine Translation Systems for WMT 16. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), volume 2, pages 464–468.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In AAAI, pages 3097–3103.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000–6010.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017. Sogou neural machine translation systems for wmt17. In Proceedings of the Second Conference on Machine Translation, pages 410–415.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

# The University of Maryland's Chinese-English Neural Machine Translation Systems at WMT18

**Weijia Xu** and **Marine Carpuat**
Department of Computer Science
University of Maryland
College Park, MD 20742, USA
weijia@cs.umd.edu, marine@cs.umd.edu

## Abstract

This paper describes the University of Maryland's submission to the WMT 2018 Chinese↔English news translation tasks. Our systems are BPE-based self-attentional Transformer networks with parallel and backtranslated monolingual training data. Using ensembling and reranking, we improve over the Transformer baseline by +1.4 BLEU for Chinese→English and +3.97 BLEU for English→Chinese on *newstest2017*. Our best systems reach BLEU scores of 24.4 for Chinese→English and 39.0 for English→Chinese on *newstest2018*.

## 1 Introduction

While machine translation between Chinese and English has long been considered a challenging task, with performance lagging behind other language pairs (Bojar et al., 2017), neural architectures have helped achieve large improvements. A new state-of-the-art on Chinese→English news translation was recently obtained (Hassan et al., 2018) using a deep Transformer model in combination with many other techniques including Dual Learning (He et al., 2016), joint training of source-to-target and target-to-source models, and Deliberation Networks (Xia et al., 2017). The resulting high quality translation comes at the cost of large models and complex training pipelines, which make such models difficult to train and deploy with constrained resources.

In this shared task, our goal is to evaluate the performance of systems inspired by Hassan et al. (2018) but with fewer and smaller components, which require less time and memory at training and decoding time. Our systems are based on a multi-layer encoder-decoder architecture with attention mechanism. We experiment with different network architectures, including single-layer RNN, deep Stacked RNN as used in Zhou et al. (2016), and self-attentional Transformer networks (Vaswani et al., 2017). The best results are obtained with deep Transformer models.

Our best systems reach BLEU scores of 24.4 for Chinese→English and 39.0 for English→Chinese on *newstest2018*. Using a combination of backtranslation (Section 2.2), ensembling, and reranking (Section 2.3) we improve over the base Transformer models by +1.4 BLEU (Chinese→English) and +3.97 BLEU (English→Chinese) on *newstest2017*. We describe each component of the system (Section 2), and its contribution for each language pair (Section 4). We show that the impact of backtranslation and reranking is not symmetric in the two translation directions, and that, compared to oracle scores, the reranker leaves much room for improvement.

## 2 Approach

### 2.1 Neural Machine Translation Models

Currently, state-of-the-art Neural Machine Translation (NMT) (Bahdanau et al., 2014) is generally based on a sequence-to-sequence encoder-decoder model with attention mechanism, which represent the conditional probability $p(y|x)$ of a target sentence $y$ given a source input $x$.

This model comprises two components: an encoder $\Theta_{enc}$ and a decoder $\Theta_{dec}$. The encoder encodes an input sentence $x$ into a sequence or set of continuous representations, while the decoder predicts the conditional probability distribution of the target words given the encoder's output states. $\Theta_{enc}$ and $\Theta_{dec}$ are trained to maximize the likelihood of a parallel training data comprised of $N$ pairs of source and target sentences:

$$\mathcal{L}(\Theta) = \sum_{n=1}^{N} \sum_{t=1}^{T} \log p(y_t^{(n)} | h_{t-1}^{(n)}, Attn; \Theta_{dec}) \tag{1}$$

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 535–540
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64058

where

$$Attn = f^{attn}(f^{enc}(x^{(n)}; \Theta enc), h_{t-1}^{(n)}) \qquad (2)$$

$h_{t-1}^{(n)}$ denotes the decoder's hidden states conditioned on $y_{<t}^{(n)}$, the target words preceding step $t$. The attention model $f^{attn}$ computes a weighted sum over the encoder's outputs $f^{enc}(x^{(n)}; \Theta enc)$ where the weights are determined by the "similarity" between each of the encoder's outputs and the decoder's hidden state $h_{t-1}^{(n)}$.

State-of-the-art NMT encoders and decoders include Stacked RNNs (Zhou et al., 2016), convolutional sequence-to-sequence models (ConvS2S) (Gehring et al., 2017), and Transformer models (Vaswani et al., 2017). The ConvS2S and Transformer models differ from RNNs in that they replace the recurrent processing in RNNs with convolutional representation and self-attention respectively, which enable the parallelization of the computation and make the encoded representation less sensitive to the sequence length.

ConvS2S uses stacked convolutional representation to model the dependencies between nearby words on lower layers, while longer-distance dependencies are modeled through upper layers. In contrast, the Transformer model captures source context via self-attention, which allows to attend to any source word regardless of position, and therefore has the potential to model long-distance dependencies more directly.

In addition, the Transformer uses multi-head attention, which lets the model attend to information from different representation subspaces at different positions. The attention function can be interpreted as mapping a query and a set of key-value pairs into an output – the output is generally computed as a weighted sum of the values, and the weights are computed by a function of the query and the corresponding key. Instead of computing a single attention pass, multi-head attention consists of several stacked attention layers in which the same attention function is applied to different transformations of the query, keys and values. And then the output vectors from the above attention layers are concatenated together and linearly transformed, resulting in the final output.

The Transformer model has achieved significant improvements over RNN-based encoder-decoders on several NMT tasks (Vaswani et al., 2017), while RNNs outperform ConvS2S (Hieber et al., 2017). We therefore only experiment with the Transformer and RNN architectures.

## 2.2 Backtranslating Monolingual Data

We leverage the monolingual data provided in the shared task using backtranslation (Sennrich et al., 2016a). For each language pair, we select monolingual corpora from the target language based on their similarity to the parallel corpus as measured by cross-entropy difference (Moore and Lewis, 2010). Following the setup from Hassan et al. (2018), we backtranslate the monolingual data using a single Transformer model, and then use a mixture of parallel and backtranslated monolingual data with a proportion of 2:1 for training a new Transformer model.

## 2.3 Reranking $n$-best Hypotheses

In order to improve the translation quality, we rerank the $n$-best results using features extracted from different NMT models (Cherry and Foster, 2012; Neubig et al., 2015; Hassan et al., 2018).

**Right-to-left NMT Model** Sequence-to-sequence models generate sequences on a token-by-token basis, and suffer from the exposure bias problem (Bengio et al., 2015). Exposure bias refers to the problem that models are trained using contexts from human generated references while tested using model-generated contexts, and thus at test time previous mistakes may be amplified and lead to subsequent errors. In order to address this issue, we train a right-to-left (R2L) NMT model using the same training data but with inverted target data. Then for each hypothesis from the $n$-best list, we invert the hypothesis sequence and use the perplexity score given by the right-to-left NMT model as a reranking feature.

**Target-to-source NMT Model** In order to improve the translation quality in terms of adequacy, we also use features from target-to-source (T2S) NMT models for reranking. We use the perplexity score given the translation as input and the source sentence as reference. The score represents the conditional probability of the source sentence given the translation, which can be viewed as an adequacy score. Since we participate in both Chinese→English and English→Chinese tasks, we can just use the models trained in the opposite direction for reranking.

**Reranking Model** First we generate $n$-best translation hypotheses for each source sentence. Then we get the perplexity scores for each hypothesis with L2R, R2L, and T2S models. The scores are treated as features which we use to train a $k$-best batch MIRA ranker (Cherry and Foster, 2012) to find out the optimal weights for reranking.

## 3 Data and Preprocessing

**Parallel Data** We use all the parallel data available for the shared tasks. The training data for both tasks consists of about 15.8M sentence pairs from the UN Parallel Corpus, 9M sentence pairs from the CWMT Corpora, 332K sentence pairs from the News Commentary Corpus. In addition to the criteria used in Hassan et al. (2018) to filter the parallel data, we add a criterion of bad sentences according to the alignment score given by the `fast-align` toolkit[1]. The overall criteria are the following:

- Duplicate sentence pairs are removed.
- Sentences with characters of other languages are removed.
- Chinese sentences without Chinese characters are removed.
- The length of each sentence must be between 3 and 50.
- The length ratio of sentence pairs must not exceed 1.6.
- Bad sentence pairs according to the alignment score are removed.

Table 1 shows the data statistics after filtering, tokenization, truecasing, and BPE.

**Monolingual Data** We further augment the training data with backtranslated monolingual data. For Chinese→English systems, we select 8M sentences from "News Crawl: articles from 2017" that are most similar to the bilingual data using cross-entropy difference (Moore and Lewis, 2010). For English→Chinese systems, we select 8M sentences from the XMU Corpus based on the same criteria.

**Tuning and Testing Data** The official newsdev2017 is used as the validation set, and newstest2017 is used as the test set.

**Preprocessing** All corpora are processed consistently. We tokenize the English sentences and perform truecasing with the Moses scripts (Koehn et al., 2007). Chinese sentences are segmented with the Jieba segmenter[2]. We segment English and Chinese tokens into subwords via Byte-pair Encoding (BPE) (Sennrich et al., 2016b). We train the BPE models for English and Chinese separately, and use 32K subwords for each side.

## 4 Experiments

### 4.1 Baseline systems

The baseline system is a bidirectional RNN with attention mechanism as used in Bahdanau et al. (2014). Our systems are built on Sockeye (Hieber et al., 2017). We use word embedding size of 1024 and hidden layer size of 1024. We filter out sentences with length larger than 50. We use Adam optimizer with initial learning rate of 0.0002. We adopt layer normalization (Ba et al., 2016) and label smoothing (Szegedy et al., 2016). We tie the output weight matrix with the target embeddings (Press and Wolf, 2017). The beam size is set to 10.

The deep RNN is based on Stacked RNNs with attention (Zhou et al., 2016). We use the same system settings as the baseline but set the number of stack layers to 4.

The Transformer network (Section 2.1) is a 6-layer Transformer model with model size of 1024, feed-forward network size of 4096, and 16 attention heads. We adopt label smoothing and weight tying, and set the beam size to 10.

Table 2 shows the total number of parameters for each model and the BLEU scores on Chinese→English and English→Chinese newstest2017. Results show that the Transformer outperforms RNNs in both directions, although it is not a controlled comparison since the Transformer has 1.6 times as many parameters as the deep RNN model. Based on this strong performance, we select the Transformer as the base model for further improvements.

### 4.2 Results on Chinese→English Translation

Table 3 shows the results for the Chinese→English translation task. We report cased BLEU computed on detokenized output with the `multi-bleu-detok.pl` script. The baseline, deep RNN, and Transformer models are trained on the 17.6M bilingual data. We backtranslate

[1] https://github.com/clab/fast_align

[2] https://github.com/fxsjy/jieba

| | **train** | | **valid** | | **test** | |
|---|---|---|---|---|---|---|
| **Sentences** | 17577153 | 17577153 | 2002 | 2002 | 2001 | 2001 |
| **Tokens** | 392490201 | 433127957 | 72494 | 69775 | 68360 | 64012 |
| **Types** | 49475 | 32102 | 4593 | 9911 | 4913 | 9171 |
| **OOVs** | – | – | 104 | 32 | 121 | 25 |

Table 1: Data sizes for Chinese/English training (train), validation (valid) and test sets respectively. All statistics are computed after filtering, tokenization, truecasing, and BPE. The *Types* column shows the number of distinct tokens in each data set. The *OOVs* column shows the number of distinct out-of-vocabulary tokens.

| **System** | **Size** | **C→E** | **E→C** |
|---|---|---|---|
| baseline | 108.77M | 20.99 | 30.45 |
| deep RNN | 165.46M | 21.65 | 31.63 |
| Transformer | 259.94M | 24.00 | 34.50 |

Table 2: BLEU scores for baseline models on Chinese→English and English→Chinese newstest2017. The *Size* column shows the total number of parameters.

| **System** | **BLEU** |
|---|---|
| baseline | 20.99 |
| deep RNN | 21.65 |
| Transformer | 24.00 |
| +synthetic | 24.12 |
| +ensemble | 24.76 |
| +reranking (L2R, T2S) | 25.20 |
| +reranking (L2R, T2S, R2L) | 25.37 |
| +beam size from 10 to 30 | **25.41** |

Table 3: Chinese → English Results on newstest2017. The submitted system is the last one.

| **System** | **BLEU** |
|---|---|
| baseline | 30.45 |
| deep RNN | 31.63 |
| Transformer | 34.50 |
| +synthetic | 36.69 |
| +ensemble | 38.28 |
| +reranking (L2R, T2S) | 38.19 |
| +reranking (L2R, T2S, R2L) | 38.42 |
| +beam size from 10 to 30 | **38.47** |

Table 4: English → Chinese Results on newstest2017. The submitted system is the last one.

the selected 8M monolingual data using the English→Chinese Transformer model. Training the Transformer model on the mixed parallel/synthetic data improves the model by +0.1 BLEU. We further train 3 independent Transformer models with different random seeds, and gain +0.6 BLEU score by ensembing. Finally, by rescoring the $n$-best lists with L2R, R2L, and T2S models, we gain +0.6 BLEU score. Increasing the beam size from 10 to 30 also brings improvements when reranking. We submit the last system and get 24.4 BLEU score on the official test set.

### 4.3 Results on English→Chinese Translation

Table 4 shows the results for the English→Chinese translation task. We report character-based BLEU calculated with the Moses `multi-bleu-detok.pl` script. Similar to the Chinese→English systems, the baseline systems are trained on the parallel data. Augmenting the training data with the backtranslated monolingual data improves BLEU by +2.2 points. The ensemble model improves over the single best system by +1.6 BLEU. Rescoring with L2R, R2L, and T2S models brings an improvement of +0.1 BLEU. We further increase the beam size from 10 to 30 to gain more from reranking. Our submitted system outperforms the best system in WMT17 (Wang et al., 2017) by +2.1 BLEU on newstest2017 and obtains a BLEU score of 39.0 on the official test set.

We note that the components added to the baseline Transformer model have an asymmetric impact in the two translation directions. While backtranslation improves the results by +2.2 BLEU for the English→Chinese task, it doesn't help as much for Chinese→English (+0.1). In contrast, rescoring with L2R, R2L, and T2S models brings more improvements for Chinese→English (+0.6) than the other (+0.2). One possible explanation is that in a parallel corpus sentences originally written in language A and sentences translated from language B to A may have different styles due to translationese effects (Volansky et al., 2015).

While the original language is not known for all training documents, it seems reasonable to assume that the majority of documents are translated from English into Chinese: the UN corpus is known to comprise primarily original English documents

| Beam | C→E |  | E→C |  |
|---|---|---|---|---|
|  | reranker | oracle | reranker | oracle |
| 10 | 25.37 | 28.72 | 38.42 | 42.84 |
| 30 | **25.41** | 30.44 | **38.47** | 44.78 |
| 100 | 25.40 | **33.05** | 38.38 | **47.17** |

Table 5: A comparison of BLEU scores when using the reranker trained with L2R, R2L, and T2S features versus the oracle, with varying beam sizes.

(Tolochinsky et al., 2018). For other training data sources beyond UN, a bilingual Chinese-English speaker manually inspected a random sample of 100 sentence pairs, and estimated that 87% sentences were originally written in English. This might explain why rescoring with the T2S models helps more in the Chinese→English direction than in the other, and why the English→Chinese systems benefit more from backtranslated data which introduces some (machine) translated Chinese to complement the translation direction observed in the parallel training data.

### 4.4 Experiments on Reranking

To estimate an upper-bound for reranking methods, we build an oracle that returns the translation in the $n$-best list that gets the highest BLEU score.

Table 5 shows the comparison of BLEU scores when using the reranker trained with L2R, R2L, and T2S features versus the oracle. Increasing the beam size from 30 to 100 doesn't improve the results when using the reranker, but improves the oracle scores. This is consistent with prior findings that beam search only improves translation quality for narrow beams and deteriorates with larger beams (Koehn and Knowles, 2017), but differs in that we rerank the $n$-best lists instead of adopting the 1-best results from beam search. The results also show that better translations according to BLEU exist in the $n$-best lists with larger beam size, but are ranked low by the models.

In addition, we find that the oracle scores are always higher than the reranker scores, and the gap increases with beam size. When comparing the MSR's best system results (28.46 BLEU achieved by Combo-4 in Hassan et al. (2018) with the oracle, we find that the oracle score is still higher by 4-5 BLEU. The results show that there is room for improvement by introducing more useful rescoring features and warrant further investigation.

## 5 Conclusion

This paper presents the University of Maryland's NMT systems for WMT 2018 Chinese↔English news translation tasks. Our experiments confirm the benefits of using Transformer networks over RNN-based architectures. We report performance gains from incorporating monolingual data, using ensemble models and reranking with target-to-source and right-to-left models, although the impact of these techniques depends on the translation direction. By comparing the oracle and reranking results, we find that there is potential for further improvement with more useful rescoring features.

## Acknowledgments

Thanks to Xing Niu for his initial scripts, and to the anonymous reviewers for their valuable comments. This work is partially funded by an AWS Machine Learning Research Award.

## References

Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *CoRR*, abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1171–1179. Curran Associates, Inc.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark.

Colin Cherry and George Foster. 2012. 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.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic Chinese to English news translation. *CoRR*, abs/1803.05567.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 820–828. Curran Associates, Inc.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *CoRR*, abs/1712.05690.

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. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden.

Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: Naist at wat2015. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 35–41.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Elad Tolochinsky, Ohad Mosafi, Ella Rabinovich, and Shuly Wintner. 2018. The UN parallel corpus annotated for translation direction. *CoRR*, abs/1805.07697.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Vered Volansky, Noam Ordan, and Shuly Wintner. 2015. On the features of translationese. *Digital Scholarship in the Humanities*, 30(1):98–118.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017. Sogou neural machine translation systems for WMT17. In *Proceedings of the Second Conference on Machine Translation*, pages 410–415.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1784–1794. Curran Associates, Inc.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4:371–383.

# EVALD Reference-Less Discourse Evaluation for WMT18

**Ondřej Bojar** **Jiří Mírovský** **Kateřina Rysová** **Magdaléna Rysová**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
`<surname>@ufal.mff.cuni.cz`

## Abstract

We present the results of automatic evaluation of discourse in machine translation (MT) outputs using the EVALD tool. EVALD was originally designed and trained to assess the quality of *human* writing, for native speakers and foreign-language learners. MT has seen a tremendous leap in translation quality at the level of sentences and it is thus interesting to see if the human-level evaluation is becoming relevant.

## 1 Introduction

The output quality of machine translation has substantially improved in the last few years thanks to the neural models (NMT). In some setups, NMT systems may even surpass the quality of human reference translations *if evaluated at the level of individual sentences*. The natural next step is (1) to start evaluating MT using larger pieces of texts, e.g. whole documents, and (2) to evaluate using methods suitable for the text quality produced by humans.

Our contribution to the WMT18 test suites responds to both of these goals. We experiment with the application of automatic, reference-less evaluation of text quality which was originally designed to evaluate texts written by humans. In this exploratory study, we do not have the human resources for a contrastive manual evaluation of the texts. We thus limit the comparison to overall MT system quality as provided by WMT.

In Section 2, we briefly describe the tool we use, EVALD. Section 3 describes the texts and MT system used. Section 4 provides and discusses the empirical results and we conclude in Section 5.

## 2 Evaluating Discourse

EVALD (Evaluator of Discourse)[1] was used for the automatic evaluation of the translated texts. There are two main versions of EVALD: EVALD for native speakers of Czech ("L1") and EVALD for non-native speakers ("L2"). The versions share the same features but differ in training texts.

EVALD L1 was trained on 1118 essays written by native speakers, while EVALD L2 was trained on 945 essays written by learners of Czech as a foreign language. Both systems use the same 180 features that can be divided into two types: (i) shallow features that use information from lower layers of language description, namely spelling, vocabulary, morphology and syntax, and (ii) deep text features directly related to surface coherence and reaching also beyond the sentence boundaries, namely coreference, discourse connectives diversity, discourse connectives quantity, and sentence information structure. Details about the systems can be found in Novák et al. (2018), Rysová et al. (2018), Novák et al. (2017), or Rysová et al. (2017).

We expect EVALD L2 to work better because it was designed and trained for evaluation of texts that are usually not fully coherent. The same aspect is expected by the automatically translated texts – they can be sometimes disrupted from the linguistic point of view.

EVALD L1 and L2 also differ in the class labels assigned. We normalize both of them to assign scores from 1 (worst) to 5 or 6 (best; L1 uses 5 classes, L2 uses 6 classes).

## 3 Data

Since the domain of WMT18 Shared Translation Task is news, we needed to find a different input

[1] `https://lindat.mff.cuni.cz/services/evald-foreign/`

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 541–545
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64059

|  | ENG | NRE | PSY | SOC | EDU | HIS | IOE | PHI | POL | CLS | ECO | LIN | NUR | BIO | CEE | MEC | PHY |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Creative Writing | 3/1 | 1/- | 1/- | 1/- |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Critique | 4/- |  | 2/2 | -/2 | 2/- | -/1 | 1/1 | 2/- | 3/- |  |  |  |  |  |  |  |  |
| Essay | 4/1 |  | -/1 | -/2 | 2/- | 1/1 |  | -/2 | -/1 | 3/- | 1/- | 1/- | 1/- |  |  |  |  |
| Proposal |  | 1/- | 1/- | -/2 | 3/- |  | 1/1 |  | 2/- |  |  | 2/1 | 1/2 | -/1 |  |  |  |
| Report |  | -/2 | -/2 | -/6 | 2/3 |  | -/3 |  | -/2 |  |  | -/3 | -/1 | 1/- | 2/- | -/1 | -/1 |
| Research Paper |  |  | 2/- | 4/- |  |  |  |  | 1/- |  | -/1 |  |  | -/2 | -/2 |  |  |

Table 1: Texts in our test suite by genre and domain. The numbers indicate texts written by a native/non-native English speaker.

texts which matches more closely to domain that EVALD is trained for.

### 3.1 Evaluated Texts

We selected Michigan Corpus of Upper-Level Student Papers (MICUSP),[2] an open-source collection of original English texts developed at the University of Michigan (English Language Institute). MICUSP contains about 830 papers (2.6 million words). The texts come from four academic areas: Humanities and Arts, Social Sciences, Biological and Health Sciences, Physical Sciences. At the same time, various text genres are present (argumentative essay, creative writing, critique/evaluation, proposal report, research paper, response paper). Authors of the papers are final year undergraduate and graduate students who reached an A grade. The corpus contains texts written by the native as well as non-native speakers of English. The overview of the MICUSP texts selected for evaluation is presented in Table 1.

The genre that should fit EVALD best is creative writing. We thus specifically extracted all 7 texts labelled as creative writing. To further extend our test suite, we selected texts of suitable length across the genres and domains, as summarized in Table 1. In total, there are 56 texts written by native speakers and 51 texts written by non-native speakers.

We segmented the texts into individual sentences and manually edited them to correct any errors in segmentation, to remove auxiliary segments like "[Figure]" and to abbreviate them occasionally by removing e.g. inline tables.

### 3.2 MT Systems Used

The final texts were included in inputs of MT systems participating in the WMT18 News Translation Task. In addition to the "primary" systems CUNI Transformer, UEDIN and the online systems, we also added three baseline (contrastive) systems: CUNI Chimera, CUNI Chimera noDepfix and CUNI Moses.

CUNI Moses is a phrase-based MT system (Koehn et al., 2007) trained on very large data and domain-adapted for the news text. CUNI Chimera (Bojar et al., 2013) is a hybrid MT system combining the outputs of transfer-based TectoMT (Žabokrtský et al., 2008) and recently also neural MT outputs from Nematus (Sennrich et al., 2017) and Neural Monkey (Helcl et al., 2018). The backbone of Chimera is nevertheless phrase-based, so Chimera suffers from the standard problems of fluency. Depfix (Rosa et al., 2012) is a rule-based grammar correction system that served very well as the last step of Chimera prior to NMT. For a contrast, we also provide the outputs of Chimera without this rule-based component.

CUNI Transformer (Popel and Bojar, 2018) is a highly optimized NMT system based on the non-recurrent architecture of Transformer (Vaswani et al., 2017). Based on the preliminary evaluation, CUNI Transformer is expected to perform comparably or better than humans when evaluating individual sentences in isolation.

UEDIN is a 4-way ensemble of deep RNN system, running left-to-right and reranked with 4 deep right-to-left systems. It uses subword units (BPE) and back-translation. The other systems are commercial ones and their description is not available.

The manual evaluation of WMT18 is still in progress, so what we can provide now are only automatic scores as reported in `matrix.statmt.org`, see Table 2. None of the WMT18 evaluations will be strictly comparable to ours due to the difference in the domain and the set of sentences. Nevertheless, it is still the best indication of MT output quality we can get.

## 4 Evaluation

We apply EVALD to all the MT outputs and also to the source. No Czech reference is available for the texts, so we take the source as the lower bound:

[2] `http://micusp.elicorpora.info/`

| System | BLEU | BLEU-cased | TER | BEER 2.0 | CharactTER |
|---|---|---|---|---|---|
| CUNI Transformer | 26.6 | 26.0 | 0.638 | 0.567 | 0.532 |
| UEDIN | 24.0 | 23.4 | 0.666 | 0.554 | 0.550 |
| CUNI Chimera noDepFix | 21.0 | 19.8 | 0.703 | 0.528 | 0.600 |
| CUNI Chimera | 20.8 | 19.2 | 0.704 | 0.522 | 0.605 |
| CUNI Moses | 17.5 | 16.4 | 0.739 | 0.509 | 0.632 |

Table 2: Automatic results of WMT18 English-Czech systems as listed at http://matrix.statmt.org/matrix/systems_list/1883.

| EVALD version | L1 | L2 |
|---|---|---|
| CUNI Transformer | 5.00±0.00 | 5.02±0.91 |
| CUNI Chimera noDepFix | 5.00±0.00 | 4.92±0.88 |
| UEDIN | 5.00±0.00 | 4.77±0.89 |
| online-B | 5.00±0.00 | 4.76±0.87 |
| CUNI Moses | 4.97±0.29 | 4.69±0.83 |
| online-A | 5.00±0.00 | 4.60±0.81 |
| CUNI Chimera | 5.00±0.00 | 4.58±0.80 |
| online-G | 4.97±0.29 | 4.58±0.81 |
| Source | 1.00±0.00 | 1.00±0.00 |

Table 3: Overall EVALD scores for individual MT systems. L1: EVALD for native speakers with 5 being the best mark, L2: EVALD for non-natives with 6 being the best possible mark.

|  | EVALD L2 Score | # Docs | # |
|---|---|---|---|
| Creative Writing | 6.00±0.00 | 7 | 56 |
| Report | 4.72±0.84 | 29 | 289 |
| Essay | 4.67±0.89 | 21 | 153 |
| Critique | 4.65±0.90 | 20 | 136 |
| Research Paper | 4.59±0.70 | 12 | 90 |
| Proposal | 4.52±0.66 | 18 | 132 |

Table 4: Results for individual genres.

|  | EVALD L2 Score | # | # Docs |
|---|---|---|---|
| HIS | 5.48±0.89 | 27 | 3 |
| ENG | 5.23±0.97 | 83 | 13 |
| NRE | 5.11±0.63 | 35 | 4 |
| IOE | 5.03±0.79 | 65 | 7 |
| PSY | 4.83±0.97 | 88 | 11 |
| SOC | 4.79±0.91 | 160 | 17 |
| BIO | 4.74±0.60 | 38 | 4 |
| CEE | 4.62±0.61 | 32 | 4 |
| ECO | 4.56±0.73 | 16 | 2 |
| EDU | 4.55±0.86 | 78 | 12 |
| POL | 4.48±0.80 | 63 | 9 |
| LIN | 4.44±0.73 | 59 | 7 |
| NUR | 4.37±0.49 | 43 | 5 |
| MEC | 4.36±0.50 | 11 | 1 |
| PHY | 4.36±0.50 | 11 | 1 |
| CLS | 4.27±0.46 | 15 | 3 |
| PHI | 4.00±0.00 | 32 | 4 |

Table 5: Results for individual domains.

|  | EVALD L2 Score | # | # Docs |
|---|---|---|---|
| Native Speaker | 4.86±0.93 | 298 | 56 |
| Non-Native Speaker | 4.68±0.82 | 558 | 51 |

Table 6: Results depending on whether the author of the English original was an English native speaker.

EVALD, trained for Czech, should very much dislike the original English text.

The overall EVALD score across the 107 texts produced by each MT system is listed in Table 3. Clearly, the L1 version of EVALD aimed at native speakers is non-discerning. All systems get almost the same score. It is actually the best possible score, but this tells us primarily that the system trained for L1 is not suitable for our setting. Only the source gets the worst possible score.

The L2 version is more interesting. As expected, English Source receives the worst rating, 1.0 with no variance at all. MT systems score around 4 or 5. While this is a clear overestimation of the text quality (6 would be the best score and e.g. phrase-based MT Moses gets 4.69), it reveals some differences between the systems.

We thus explore only EVALD L2 in the following.

Table 4 lists EVALD L2 scores for individual genres across MT systems; Source was not considered. The columns "#" and "# Docs" specify the size of the sample in terms of individual scorings and distinct documents, respectively.

We see that all 56 translations of the 7 documents of Creative Writing seemed excellent. Again, EVALD is non-discerning in this setting. Other genres exhibit some divergence in scores. Since all the genres differ from the news texts that the MT systems are geared towards, it is not easy to explain the stability of the score in Creative Writing. Possibly, EVALD is checking many shallow discourse features (e.g. the presence of a certain variety of conjunctions) and our texts in Creative Writing superficially include the required diversity, and this diversity is preserved by all MT systems.

Table 5 looks at text domains. There is a reasonable variance across the translations and texts (except PHIlosophy) but it is again difficult to come up with a unified view. For instance, natural sciences like BIOlogy or PHYsics span a wide range

|  | Discourse-Specific | Other | All |
|---|---|---|---|
| CUNI Transformer | 4.56±1.18 | 4.79±1.16 | 5.02±0.91 |
| CUNI Chimera noDepFix | 4.52±1.15 | ≀ 4.86±1.17 | 4.92±0.88 |
| UEDIN | 4.52±1.15 | 4.86±1.09 | 4.77±0.89 |
| online-B | 4.39±1.17 | 4.82±1.12 | 4.76±0.87 |
| online-G | 4.35±1.15 | 4.68±1.21 | 4.58±0.81 |
| online-A | 4.34±1.12 | 4.66±1.28 | ≀ 4.60±0.81 |
| CUNI Moses | 4.30±1.24 | ≀ 4.69±1.28 | ≀ 4.69±0.83 |
| CUNI Chimera | 3.98±1.36 | 4.66±1.20 | 4.58±0.80 |
| Source | 1.86±1.65 | 2.00±1.73 | 1.00±0.00 |

Table 7: Comparison of EVALD L2 scores using discourse-specific (deep) features, other (shallow) features, and all features. Vertical tildes mark differences in rank in comparison with the rank given by the discourse-specific features.

|  | Avg. var. of scores |
|---|---|
| across nativeness | 0.88 |
| across MT systems | 0.85 |
| across genre | 0.67 |
| across domain | 0.67 |

Table 8: Variance in EVALD L2 scores across various aspects of our test suite.

of ranks, as humanities do (HIStory or the mentioned PHIlosophy).

Table 6 documents the effect of the mother tongue of the author of the original English text before the translation.

Table 7 compares EVALD L2 scores in three experimental settings: using only the deep text features (marked discourse-specific in the table), shallow features (marked other) and all features.[3] Vertical tildes mark differences in rank in comparison with the rank given by the deep text features. Agreement in five first ranks using the deep features and all features indicates that the full version of EVALD (i.e. using all features) really evaluates the translation systems based on the quality of the text coherence, rather than on the basis of shallow features.

Table 8 summarizes the variance of EVALD scores according to individual aspects captured in the previously mentioned tables. The highest variance of the scores appeared in the aspect of nativeness of the text author.

The second most diverse results are across MT systems. The evaluation proposed here thus seems as a promising research direction, although a careful analysis of EVALD features and their adaptation will be needed to obtain more discerning evaluation. Finally, the genre and domain of the original text also play a role but this is always to be expected.

## 5 Conclusion

We presented the results of automatic evaluation of Czech text quality applied to the output of generally good MT systems translating from English into Czech.

The results indicate that EVALD, as now trained for human-authored texts, is ineffective in its version for native speakers. However, EVALD version for non-natives has a rather promising potential for evaluating automatic translations because it allows distinguishing individual MT systems.

The most diversity of scores can be attributed to the nativeness of the author of the original text. We conclude that the examined MT systems in general preserve sufficient traits of source text quality for this.

EVALD-style of evaluation seems promising because the second most differentiating aspect is the MT system used. Further exploration of EVALD features as well as a direct comparison with manual assessment of translation quality are, however, necessary to make EVALD a useful MT evaluation method.

## Acknowledgments

This work has been supported by the grants 18-24210S, GA17-06123S and GA17-03461S of the Czech Science Foundation, SVV 260 453, and using language resources and tools distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (projects LM2015071 and OP VVV VI CZ.02.1.01/0.0/0.0/16 013/0001781).

## References

Ondřej Bojar, Rudolf Rosa, and Aleš Tamchyna. 2013. Chimera – Three Heads for English-to-Czech Translation. In *Proceedings of the Eighth Workshop on*

[3] See Section 2 for the list of features.

*Statistical Machine Translation*, pages 92–98, Sofia, Bulgaria. Association for Computational Linguistics.

Jindřich Helcl, Jindřich Libovický, Tom Kocmi, Tomáš Musil, Ondřej Cífka, Dušan Variš, and Ondřej Bojar. 2018. Neural monkey: The current state and beyond. In *The 13th Conference of The Association for Machine Translation in the Americas, Vol. 1: MT Researchers' Track*, pages 168–176, Stroudsburg, PA, USA. The Association for Machine Translation in the Americas, The Association for Machine Translation in the Americas.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Michal Novák, Jiří Mírovský, Kateřina Rysová, and Magdaléna Rysová. 2018. Topic-focus articulation: A third pillar of automatic evaluation of text coherence. conference paper. In *Proceedings of MICAI 2018*. Springer LNAI – Lecture Notes in Artificial Intelligence, in press.

Michal Novák, Kateřina Rysová, Magdaléna Rysová, and Jiří Mírovský. 2017. Incorporating coreference to automatic evaluation of coherence in essays. In *Statistical Language and Speech Processing*, number 10583 in Lecture Notes in Computer Science, pages 58–69, Cham, Switzerland. Claude Chappe Informatics Institute at University of Le Mans, Springer International Publishing.

Martin Popel and Ondej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Rudolf Rosa, David Mareček, and Ondej Dušek. 2012. DEPFIX: A System for Automatic Correction of Czech MT Outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 362–368, Montréal, Canada. Association for Computational Linguistics.

Kateřina Rysová, Magdaléna Rysová, Jiří Mírovský, and Michal Novák. 2017. Introducing EVALD software applications for automatic evaluation of discourse in czech. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 634–641, Šumen, Bulgaria. Bulgarian Academy of Sciences, INCOMA Ltd.

Magdaléna Rysová, Kateřina Rysová, Jiří Mírovský, and Michal Novák. 2018. Practicing students writing skills through elearning: Automated evaluation of text coherence in czech. In *EDULEARN18 Proceedings*, pages 1963–1970, Valencia, Spain. IATED Academy, IATED Academy.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. TectoMT: Highly Modular Hybrid MT System with Tectogrammatics Used as Transfer Layer. In *Proc. of the ACL Workshop on Statistical Machine Translation*, pages 167–170, Columbus, Ohio, USA.

# The WMT18 Morpheval test suites for English–Czech, English–German, English–Finnish and Turkish–English

**Franck Burlot**
Lingua Custodia
1, Place Charles de Gaulle
78180 Montigny-le-Bretonneux
franck.burlot@linguacustodia.com

**Yves Scherrer**
Department of Digital Humanities
University of Helsinki
Helsinki, Finland
yves.scherrer@helsinki.fi

**Vinit Ravishankar**
Charles University
Faculty of Mathematics and Physics
ÚFAL; Prague, Czech Republic
vinit.ravishankar@gmail.com

**Ondřej Bojar**
Charles University
Faculty of Mathematics and Physics
ÚFAL; Prague, Czech Republic
bojar@ufal.mff.cuni.cz

**Stig-Arne Grönroos**
Aalto University
Department of Signal
Processing and Acoustics
Espoo, Finland
stig-arne.gronroos@aalto.fi

**Maarit Koponen**
School of Languages and
Translation Studies
University of Turku
Turku, Finland
maarit.koponen@utu.fi

**Tommi Nieminen**
Department of Digital Humanities
University of Helsinki
Helsinki, Finland
tommi.nieminen@helsinki.fi

**François Yvon**
LIMSI, CNRS, Université Paris Saclay
Campus Universitaire d'Orsay
F-91 403 Orsay Cédex
francois.yvon@limsi.fr

## Abstract

Progress in the quality of machine translation output calls for new automatic evaluation procedures and metrics. In this paper, we extend the Morpheval protocol introduced by Burlot and Yvon (2017) for the English-to-Czech and English-to-Latvian translation directions to three additional language pairs, and report its use to analyze the results of WMT 2018's participants for these language pairs. Considering additional, typologically varied source and target languages also enables us to draw some generalizations regarding this morphology-oriented evaluation procedure.

## 1 Introduction

The success of rather opaque neural machine translation systems has called for more fine-grained types of evaluation than traditional automatic evaluation metrics offer. In particular, we would like to obtain more detailed information about systems performance than just one overall number (even if it correlates well with human judgement). Evaluation metrics that focus on various aspects of the translation, such as syntax or morphology, rather than on general translation quality, have thus seen renewed interest. This interest has spurred the inclusion of additional test suites into the WMT 2018 news translation task.

Burlot and Yvon (2017, B&Y in the following) present a test suite for evaluating the morphological competence of machine translation systems. They provide a set of sentence pairs in the source language that differ by one morphological contrast. A sentence pair is considered correct if the morphological contrast is also conveyed in the target language translations of the two sentences of the pair. B&Y developed their test suite for English–Czech and English–Latvian and applied it to a selection of MT systems that participated in WMT 2017. For WMT 2018, we have

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 546–560
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64060

extended the English–Czech test suite[1] and created similar Morpheval test suites for three additional translation directions: English–German,[2] English–Finnish,[3] and Turkish–English.[4] All primary WMT submissions of these translation directions were evaluated.[5]

We start by summarizing the components of the Morpheval test suites and their language-specific implementations.

## 2 The Morpheval test suites

A Morpheval test suite according to B&Y consists of three aspects:

- the definition of a set of contrasts that can be triggered in the source language and evaluated in the target language;
- a procedure to generate contrast pairs from a monolingual source language corpus;
- and a procedure to score the target language translations of the contrast pairs.

B&Y describe three types of contrasts. Type A contrasts resemble paradigm completion tasks, in which one single morphological feature (number, gender, tense, etc.) is evaluated. The two sentences of a contrast pair only differ in one word (or phrase) and across one feature at a time. Type B contrasts contain somewhat more complicated substitutions that are mainly evaluated in terms of agreement. For example, a contrast pair contains a pronoun or an adjective-noun noun phrase, and its evaluation is correct if the adjective and noun agree. Type C contrasts concern lexical replacements of the same category, testing whether the morphological agreement still holds if an adjective is replaced by a hyponym. Table 1 summarizes the set of contrasts implemented for the different language pairs, according to this typology. The contrasts that are not described in B&Y will be presented in detail in the following sections.

Before that, we discuss some language-specific implementation differences of the generation and scoring procedures.

### 2.1 Sentence selection and contrast generation

We follow the algorithm provided by B&Y for sentence selection and contrast generation:

1. Collect a large number of short sentences (length $<$ 15 words) containing a source feature of interest.

   As source corpora, we use the English News-2007 and 2008 corpora (for EN-CS and EN-DE), the English News-2007 corpus (for EN-FI), and SETIMES2 (for TR-EN). In order to detect the source features, the corpora are annotated using TreeTagger (Schmid, 1994) and/or CoreNLP (Manning et al., 2014) (for English), or an Apertium (Forcada et al., 2011) morphological analyser (for Turkish). For the named entities feature used in EN-FI, we additionally annotate the source corpora with the Stanford NER tagger (Finkel et al., 2005).

2. Generate a variant as prescribed by the contrast feature.

   For English corpora, we follow B&Y and use the Pymorphy morphological generator[6] to create the variants. For the Turkish corpus, we use Apertium.

3. Compute an average language model (LM) score for the base/variant pair, and remove the 33% worst pairs based on the LM score.

   We use a 5-gram language model trained on all English monolingual data available at WMT 2015. No language model filtering is applied to the Turkish data.

4. Randomly select 500 pairs per feature (400 for Turkish) for inclusion.

B&Y identify one of the sentences of a contrast pair as the "base" and the other one as the "variant". We keep this terminology for the sake of simplicity, but do not intend to imply (1) that the base is in any way "easier" to translate than the

[1] Contributors: Franck Burlot and François Yvon; test suite and evaluation scripts are available at https://github.com/franckbrl/morpheval_v2

[2] Contributors: Franck Burlot and François Yvon; test suite and evaluation scripts are available at https://github.com/franckbrl/morpheval_v2

[3] Contributors: Yves Scherrer, Maarit Koponen, Tommi Nieminen, Stig-Arne Grönroos; test suite, evaluation scripts and logs are available at https://github.com/Helsinki-NLP/en-fi-testsuite

[4] Contributors: Vinit Ravishankar and Ondřej Bojar

[5] The same method has also been adapted to English-to-French: significance tests, as well as concrete examples, are provided for this language pair in Burlot and Yvon (2018).

[6] http://pymorphy.readthedocs.io/

| Feature | B&Y | EN-CS | EN-DE | EN-FI | TR-EN |
|---|---|---|---|---|---|
| **Paradigm contrast features:** | | | | | |
| Singular vs. plural noun | A-1 | ✓ | ✓ | ✓ | |
| Singular vs. plural pronoun | A-2 | ✓ | ✓ | ✓ | |
| Masculine vs. feminine pronoun | A-3 | ✓ | ✓ | S | |
| Present vs. future tense | A-4 | ✓ | ✓ | S | ✓ |
| Present vs. past tense | A-5 | ✓ | ✓ | ✓ | ✓ |
| Indicative vs. conditional mode | | ✓ | ✓ | | |
| Positive vs. comparative adjective | A-6 | ✓ | ✓ | ✓ | |
| Positive vs. superlative adjective | | ✓ | ✓ | | |
| Affirmative vs. negative verb form | A-7 | ✓ | ✓ | ✓ | ✓ |
| Compound generation | | | ✓ | | |
| Human vs. non-human pronoun | | | | ✓ | |
| Definite vs. possessive determiner | | | | ✓ | |
| Definite vs. indefinite determiner | | | | S | |
| Reported speech subordinate clauses | | | | ✓ | |
| First vs. second person verb form | | | | | ✓ |
| Present vs. future subject participle | | | | | ✓ |
| Present vs. future object participle | | | | | ✓ |
| **Agreement features:** | | | | | |
| Pronoun vs. Adj+Nouns | B-1 | ✓ | ✓ | ✓ | |
| Pronoun vs. coordinated nouns | B-2 | ✓ | | | |
| Simple vs. coordinated verbs | B-3 | ✓ | ✓ | | |
| Adposition case (+ position) | B-4 | ✓ | ✓ | ✓ | |
| Coreference link | | ✓ | ✓ | | |
| Strong/weak adjective | | | ✓ | | |
| Local postposition/adverb case | | | | ✓ | |
| **Rare word features:** | | | | | |
| Named entities | | | | ✓ | |
| Numbers | | | | ✓ | |
| **Consistency features:** | | | | | |
| Adjective hyponyms | C-1 | ✓ | ✓ | | |
| Noun hyponyms | C-2 | ✓ | ✓ | | |
| Verb hyponyms | C-3 | ✓ | ✓ | | |

Table 1: List of contrast features implemented in the Morpheval test suites. The features already proposed by B&Y are marked by their corresponding code in the second column. *S* indicates features used to measure stability (see Section 2.5).

variant, (2) that the base always is the unmodified sentence extracted from the corpus and the variant the automatically modified one, or (3) that the evaluation of the base would be more lenient than the evaluation of the variant.

For consistency features (see Table 1), we select a noun, an adjective or a verb and replace it with a random hyponym, producing an arbitrary number of sentences. Sentence selection slightly differs from the description above: during step 2, we generate as many variants as possible. Each variant is then scored with a language model and only the top four variants are kept, leading to buckets of five sentences. For hyponym generation, we use WordNet (Miller, 1995).

### 2.2 Scoring procedures

The automatic scoring procedure for a given contrast pair receives two target language sentences (the MT output of the two source language sentences forming the contrast pair) as input and returns a binary correct/incorrect judgement. A contrast pair is judged correct if the two target sentences differ and the differences encode the contrast that is expressed in the source sentences. A contrast pair is judged incorrect if the two sentences are identical or if they differ in a way that is irrelevant to the examined contrast.

For consistency features, we wish to assess the MT system consistency with respect to lexical variation in a fixed context; accordingly, we measure the success based on the average normalized entropy of morphological features in the set of target sentences.

The target language sentences of all participating systems are morphologically analyzed to facilitate scoring. The following tools are used:

- Czech: MorphoDiTa (Straková et al., 2014)
- German: SMOR (Schmid et al., 2004)
- Finnish: The *finnish-analyze-words* script[7] provided by the Language Bank of Finland[8] and based on the Omorfi morphology (Pirinen, 2015) and the HFST toolkit (Lindén et al., 2011)
- English: MorphoDiTa (Straková et al., 2014)

As shown by B&Y, there is no need to perform a full morphological disambiguation in the target side, as we merely need to check whether some morphological features are present or absent. In fact, full automatic disambiguation could be harmful due to error propagation.

### 2.3 Additional English–Czech contrasts

The English–Czech evaluation procedure follows B&Y, to which we added a handful of new tests.

#### Conditional

Paradigm contrast features introduce a new verbal test. In the test suite, a verb in future tense is turned into its conditional form: *I will write* → *I would write*. In the Czech variant, we check whether the verb translation is in conditional mode.

#### Superlative

The superlative task is comparable to the comparative task introduced in B&Y. The base sentence contains an adjective and the variant contains its superlative form. In the output, we look for the adjective translation and check whether is has a superlative form.

#### Coreference

Agreement features introduce a new coreference task. The test suite for this task was produced using English coreference annotations obtained using CoreNLP (Manning et al., 2014). We collected sentences containing a coreference link involving a personal pronoun (*it*) or a relative pronoun (*that, which, who, whom, whose*). The base sentence remains unchanged. In order to generate the variant, the antecedent noun of the pronoun is then changed to a synonym using WordNet (Miller, 1995):

- Personal pronoun: *This **cat** is cute and I love **it**.* → *This **dog** is cute and I love **it**.*
- Relative pronoun: *The **woman who** left was angry.* → *The **man who** left was angry.*

In the output of the MT system, we are then able to locate the antecedent of the pronoun by looking for the only noun that differs between the base and variant translations (namely, the translation of *cat/woman* in the base and *dog/man* in the variant). Finally, we check whether the noun and personal

[7] http://urn.fi/urn:nbn:fi:lb-2018041701

[8] https://www.kielipankki.fi/

pronoun bear the same gender.[9] We also check number agreement for the relative pronoun. Note that for this specific task, we can compute accuracy scores on both base and variant.

### 2.4 Additional English–German contrasts

English–German is a new language pair we introduce in the current paper. It takes most of the previous tasks introduced in B&Y for English into Czech and Latvian. Conditional, superlative and coreference tasks are also adapted to German (see Section 2.3).

#### Compounds

This task consists in assessing the ability of the MT system to generate correct compounds that actually exist in German. For this purpose, the base sentence in the English test suite contains a multi-word expression that is *most likely* translated by a compound in German. To generate the variant, we modify one single English word in the multi-word expression, such that the new German translation should result in a compound that has at least one morpheme in common with the one seen in the base translation. For instance, the English expression *apple juice* in the base translates into the German compound *Apfelsaft*. We modify the word *apple* and obtain *orange juice*, which translates into *Orangensaft*. In the MT output we finally compare both compounds *Apfelsaft* and *Orangensaft* and report a success if they have at least one morpheme in common. Here, the common morpheme is *-saft*.

For the test suite generation, we needed a translation dictionary containing compounds on the German side and multi-word expressions on the English side. We gathered all the English-German parallel data we could find on OPUS (Tiedemann, 2012) and removed the data available at the WMT18 News Translation shared task. This resulted in nearly 40M parallel sentences. We obtained a phrase table out of this data using the Moses toolkit (Koehn et al., 2007). We finally extracted from this phrase table a dictionary containing a compound on the German side and several multi-word expressions on the English side (removing punctuation and other noisy tokens).

The test suite generation starts with the identification in the base sentence of an English multi-word expression that is present in our dictionary. We then look for a new English multi-word expression that has at least one common word with the previous one (we have *apple juice*, we get the expression *orange juice*, since both have *juice* in common). Finally, if both expressions translate into German compounds that have at least one morpheme in common (relying on SMOR analysis), the new English expression is inserted into the sentence, which produces the variant sentence.

At evaluation step, we look for the word in the base sentence that is not in the variant sentence and vice-versa. We report a success when both words are known compounds and when they contain at least one common morpheme (using SMOR analysis).

#### Verb position

The test suite is generated by locating complex sentences where (a) the principal clause can be omitted and (b) the subordinate clause leads to a German translation where the verb should be located at the end of the clause. Using CoreNLP annotations, we focus on specific English conjunctions that lead to a verb shift in German, like *that* → *dass*, *because* → *weil*, etc. In order to generate the variant sentence, we simply omit all words from the beginning of the sentence up to the conjunction: *I think* ***that*** *life is hard.* → *Life is hard.*

Once both sentences are translated into German, we simply check that the conjugated verb is closer to the end of the sentence in the base than it is in the variant: *Ich denke, dass das Leben hart* ***ist****.* (last position) → *Das Leben* ***ist*** *hart.* (second to last).

#### Strong adjective

This task focuses on the contrast between weak and strong forms of the German adjective. We rely on a quite simple rule of German, stating that an adjective following a definite article does not contain any gender marker in its ending, whereas it does contain it when following, e.g. a possessive determiner.

We therefore identified English sentences with a subject noun phrase containing a definite article, an adjective and a noun (according to CoreNLP analysis). To generate the variant, we simply replace the article by a possessive determiner: *The small dog is gone.* → *Our small dog is gone.*

[9] We do not control whether the chosen synonym translates to a noun of a different gender. In some cases, the translation of the pronoun in the base and variant should thus remain unchanged, sharing the same gender.

In the MT output, we check whether the variant contains a strong form of the adjective (using SMOR analysis): *Der kleine Hund ist weg.* → *Unser **kleiner** Hund ist weg.*

## 2.5 Additional English–Finnish contrasts

For English–Finnish, we reuse most of B&Y's paradigm contrast features, but repurpose some of them as stability features (see Table 1 and below). We reuse a limited subset of agreement features. After initial experiments, we decided against using consistency features, as they yielded a high percentage of unnatural and sometimes even unintelligible sentences. We provide additional features tailored to Finnish in both categories and provide an additional class of language-independent rare word features. In the following sections, we describe these features in more detail.

### Human vs. non-human pronoun

Both English and Finnish distinguish between pronouns whose antecedents are human (English *I, he, she,* ..., Finnish *minä, hän,* ...) and pronouns whose antecedents are non-human (English *it*, Finnish *se*).

The conversion procedure identifies base sentences with instances of *me, us, him*, or *her*, and generates the variants by replacing the pronouns with *it*. We discard subject contexts and make sure that no other pronoun is present in the sentence. We also discard prepositional phrase contexts which would command the use of possessive suffixes in Finnish. Note that no treatment is applied to the antecedent of the pronoun. This is generally not an issue because we do not need to preserve the meaning between the base and variant sentence, we only need to check if human vs. non-human aspect of the pronoun is preserved.

The scoring procedure checks if the correct Finnish pronoun lemma (*se*) is used in the variant.

### Definite vs. possessive determiner

In contrast to English, Finnish uses suffixation to indicate possession, e.g. *-ni* for the 1st person singular and *-si* for 2nd person singular as in *kirja+ni* 'my book', *kirja+si* 'your book'. We wanted to test how well current MT systems are able to generate these suffixes.

The conversion procedure selects variant sentences with noun phrases containing a possessive determiner and generates the base by replacing the possessive determiner with *the*.

The scoring script checks whether the possessive suffix (or alternatively, the possessive determiner) of the correct person is generated.

### Reported speech subordinate clauses

In English, the structure of affirmative and interrogative subordinate clauses is rather similar: *X says that A* vs. *X asks if A*, without any structural differences in X or A. In Finnish, various types of expressions A are possible for *say+that*, but none of them is structurally identical to the *ask+if* subordinate clause, which corresponds to a (direct) question with the question particle *-ko/kö*.

The conversion procedure is bidirectional: it selects sentences containing *say+that* and transforms them to *ask+if* and vice-versa. Idiomatic constructions like *having said that* or *when asked if* are discarded.

The scoring procedure reports success if one of the correct constructions is identified in the affirmative sentence, and if the *-ko/kö*-construction is identified in the interrogative sentence.

### Stability features

Two of the paradigm contrast features reported by B&Y do not apply to Finnish. Feature A-3 tests whether the masculine/feminine contrast between the pronouns *he* and *she* is conveyed in the target language, but Finnish uses the same pronoun *hän* regardless of the gender of the antecedent. Feature A-4 tests whether the present tense/future tense contrast is conveyed in the target language, but Finnish does not have a future tense and generally uses present tense in such cases.

Instead of measuring contrast, we can use these two features to measure *stability*: an MT system can be considered stable if two source sentences differing only in one word according to the contrasts presented above yield completely identical translations. Note that stability is not necessarily a good measure of overall translation quality: text can be translated in various ways, and two completely different translations can still be both correct, adequate and natural. However, stability may be an important criterion for particular applications of machine translation. For instance, for purposes of manual post-editing, stability may be preferable as it leads to easier predictability of the output. Our findings concerning the relation between stability and general translation quality will be discussed below.

We introduce a third stability feature that relies on the absence of determiners in Finnish: we select sentences with noun phrases containing the indefinite determiner *a* and replace it with the definite determiner *the*. We try to avoid noun phrases in object positions, where determinacy can be expressed through case in Finnish.

The scoring procedure for stability features is simple: a contrast pair is considered stable if the strings of both translations are identical.

These stability features can be compared to the consistency features used for Czech and German. For both feature types, the variants are created through some type of transformation that is supposed to be invariant with respect to target morphology. For the consistency features, this transformation is semantic (based on the hyponymy relation), whereas it is morphological for the stability features.

### Adposition case

B&Y introduce a feature where an English preposition is replaced by another one such that their counterparts in the target language govern two different cases. In Finnish, case government is closely tied to word order: most adpositions are postpositions and require genitive case, but some adpositions are prepositions and require partitive case. There are only two frequent prepositions, namely *ennen* 'before' and *ilman* 'without'. We restrict this feature to the former, as the latter often appears in idiomatic expressions from which variants are difficult to generate.

The sentence selection script produces the contrast pairs *before* → *after* and *before* → *during* (base followed by variant). Idiomatic constructions such as *named after, looking after, come before* are discarded, as well as particle readings of these words.

The scoring procedure verifies if a preposition with a noun or pronoun in partitive case to its right is present in the base, and if a postposition with a noun or pronoun in genitive case to its left is present in the variant. It also accepts the postpositional use of *ennen* in conjunction with pronouns (*sitä ennen*), as well as the use of bare (ad-/in-)essive case instead of the postposition *aikana* 'during'.

### Local postposition case

Finnish local postpositions (the equivalents of *over, under, next to, between*, etc.) can be inflected themselves using the Finnish local cases, e.g. *sisällä/sisältä/sisälle* 'inside/from inside/towards inside', *edessä/edestä/eteen* 'in front of/from in front of/towards in front of'.

The conversion procedure yields the following contrast pairs: *in front of* → *behind*, *underneath* → *next to*, *outside* → *inside*, *inside* → *outside*, *above* → *below*, *below* → *above*. Non-prepositional and idiomatic readings are discarded as far as they could be discovered during development.

The scoring procedure checks that the English prepositions are translated correctly and that the case type (locative/separative/lative, as in the examples above) matches between the two sentences of the contrast pair.

### Rare word features

In the early days of NMT, translation of out-of-vocabulary words was virtually impossible and hampered the performance when compared with SMT. In recent years however, most systems have adopted an approach in which rare words are split into "subwords" during preprocessing (see e.g. Sennrich et al., 2016), such that any unknown word can be composed of various subword chunks during test time. Several subword chunking algorithms with various parameter settings can be used, but their respective performance differences are hard to assess as they typically concern low-frequency words with low impact on general translation quality. Therefore, we introduce two features that specifically deal with low-frequency items. These features are language-independent and do not require the use of a morphological analyzer.

For the first feature, we identify large numbers (at least 3 digits) in the English source text and modify them by subtracting a constant number. For example, the number *27,801* would be transformed into *27,628*. The scoring procedure verifies if the original and modified numbers are found in the respective sentences.

For the second feature, we use the Stanford Named entity recognizer to identify named entities in the English source text. We then consider two subsets of named entities, frequent ones (occurring more than 1000 times) and rare ones (occurring between 20 and 100 times). Contrast pairs are generated by identifying sentences with a frequent named entity, and replacing it by a rare one. We restrict the replacement to single-word named entities of the same class and make sure that the replacement candidate contains at least two differing

| System | BLEU | Ave. z |
|---|---|---|
| CUNI-Transformer | 26.6 | 0.594 |
| uedin | 24.0 | 0.384 |
| online-B | — | 0.101 |
| online-A | — | -0.115 |
| online-G | — | -0.246 |

Table 2: BLEU scores and human evaluation scores computed on newstest-2018 for English–Czech.

| System | BLEU | Ave. z |
|---|---|---|
| online-Z | — | 0.653 |
| online-B | — | 0.561 |
| Microsoft-Marian | 48.9 | 0.551 |
| MMT-production-system | 46.7 | 0.539 |
| UCAM | 47.1 | 0.537 |
| NTT | 47.0 | 0.491 |
| KIT | 46.9 | 0.454 |
| online-Y | — | 0.396 |
| JHU | 43.9 | 0.377 |
| uedin | 44.9 | 0.352 |
| LMU-nmt | 40.6 | 0.213 |
| online-A | — | 0.060 |
| online-F | — | -0.385 |
| online-G | — | -0.416 |
| RWTH-UNSUPER | 15.9 | -0.966 |
| LMU-unsup | 15.8 | -1.122 |

Table 3: BLEU scores and human evaluation scores computed on newstest-2018 for English–German.

characters, as in the following example: *Extensive damage was reported in Cuba. → Extensive damage was reported in Tuzla.*

The scoring procedure checks that both named entity strings are found in the respective sentences. The frequent named entities are likely to be translated (e.g., English *Africa* would become Finnish *Afrikka*, in oblique cases *Afrika-*). Therefore, we add a small hand-crafted dictionary containing the most frequent entities, and compare these entries with the base forms obtained by the morphological analyzer. We currently do not verify case consistency, as many rare entities are not recognized by the morphological analyzer.

### 2.6 Additional Turkish–English contrasts

We introduce Turkish–English as another new language pair in the paper. Note that the translation direction is opposite to the other pairs, with English acting as the target language. We include the B&Y tests for verb tense and polarity and add several tests for Turkish-specific features.

| System | BLEU | Ave. z |
|---|---|---|
| NICT | 18.2 | 0.521 |
| HY-NMT | 17.8 | 0.466 |
| uedin | 16.7 | 0.324 |
| Aalto | 16.2 | 0.271 |
| HY-NMT2step | 14.5 | 0.258 |
| talp-upc | 14.3 | 0.238 |
| CUNI-Kocmi | 14.7 | 0.184 |
| online-B | — | 0.183 |
| online-A | — | -0.212 |
| online-G | — | -0.233 |
| HY-SMT | 10.5 | -0.334 |
| HY-AH | 6.4 | -0.369 |

Table 4: BLEU scores and human evaluation scores computed on newstest-2018 for English–Finnish.

|  | SETIMES2 | newstest-2018 |  |
|---|---|---|---|
| System | BLEU | BLEU | Ave.z |
| online-G | 25.86 | — | 0.101 |
| online-A | 27.03 | — | 0.077 |
| Alibaba-Ensemble | — | — | 0.030 |
| online-B | 24.84 | — | 0.027 |
| uedin | 48.42 | 26.9 | -0.008 |
| NICT | 40.64 | 26.7 | -0.040 |

Table 5: BLEU scores computed on SETIMES2 and newstest-2018 and human evaluation scores on newstest-2018 for Turkish–English.

#### Verb person

Turkish models verbal agreement with number and person agglutinatively, often making pronouns superfluous. We modify first person verbal agreement to second person, keeping the number intact: *kitap okuyorum → kitap okuyorsun*. We check the MT output for the presence of the pronoun *you*: *I am reading a book → **you** are reading a book*.

#### Participles

Turkish features several participles that form relative clauses. These include, relevant to our tests, present-tense subject and object participles, and future tense participles. We introduce two tests. One transforms present tense subject participles to future tense ones: *Bu gelen adam → Bu gelecek adam*. For the English translations, our (fairly simple) test involves searching through the translation output for the tense-imparting strings, *will, shall, would* and *going* (as a simple test for the presence of *'going to'*): *The man who is coming → The man who **will** come*.

Object participles function similarly, however, they use transitive verbs that take an argument: *Okuduğum kitap → okuyacağım kitap*. Our tests for the MT output are similar: *the book that I read*

| | Verbs | | | | Pronouns | | Nouns | Adjectives | | Average |
|---|---|---|---|---|---|---|---|---|---|---|
| System | Past | Future | Cond. | Neg. | Fem. | Plur. | Plur. | Compar. | Superl. | |
| CUNI-Transformer | 84.2 | **88.0** | 59.0 | **97.4** | 94.2 | 92.2 | 76.4 | 74.0 | **89.8** | 83.9 |
| uedin | **92.0** | 83.0 | **73.4** | 96.6 | 94.2 | **92.8** | 78.8 | **78.8** | 88.8 | **86.5** |
| online-B | 87.8 | 77.6 | 57.4 | 94.2 | 92.8 | 92.0 | 80.0 | 75.8 | 69.8 | 80.8 |
| online-A | 86.8 | 86.8 | 71.2 | 94.4 | 94.0 | 89.6 | **81.2** | 74.6 | 61.0 | 82.2 |
| online-G | 81.4 | 84.0 | 70.8 | 78.4 | **98.0** | 89.4 | 79.2 | 73.0 | 50.4 | 78.3 |

Table 6: Accuracy values for the English-Czech test suite (paradigm contrast features).

| | Coordinated verbs | | | Co. N. | Adj+Nouns | | | Coref. rel. | | Prep. | Cor. per. | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| System | Nbr | Pers | Tense | Case | Gdr | Nbr | Case | Gdr | Nbr | Case | Gdr | |
| CUNI-Transformer | **88.0** | **88.4** | **84.8** | 99.6 | **96.4** | **97.0** | **97.0** | 76.5 | 77.3 | 95.4 | **66.4** | **87.9** |
| uedin | 84.8 | 84.8 | 81.8 | 99.8 | 94.6 | 94.8 | 94.8 | 81.0 | 82.2 | **96.9** | 64.3 | 87.3 |
| online-B | 82.2 | 83.2 | 80.0 | 99.8 | 91.2 | 91.8 | 91.6 | 79.5 | 80.1 | 91.6 | 64.8 | 85.1 |
| online-A | 68.8 | 67.6 | 65.0 | 99.2 | 91.0 | 89.4 | 91.2 | **81.9** | **82.9** | 96.1 | 55.0 | 80.7 |
| online-G | 62.6 | 61.2 | 58.8 | **100.0** | 83.2 | 80.4 | 82.6 | 76.7 | 77.5 | 84.6 | 42.0 | 73.6 |

Table 7: Accuracy values for the English-Czech test suite (agreement features).

| | Nouns | Adjectives | | | Verbs | | | | Average |
|---|---|---|---|---|---|---|---|---|---|
| System | Case | Gender | Number | Case | Number | Person | Tense | Negation | |
| CUNI-Transformer | 0.109 | 0.191 | 0.193 | 0.203 | 0.110 | 0.077 | **0.096** | 0.069 | 0.131 |
| uedin | **0.095** | **0.185** | **0.184** | **0.189** | **0.099** | 0.081 | 0.097 | 0.072 | **0.125** |
| online-B | 0.105 | 0.186 | 0.186 | 0.195 | 0.108 | **0.071** | 0.099 | **0.067** | 0.127 |
| online-A | 0.096 | 0.202 | 0.201 | 0.207 | 0.182 | 0.129 | 0.154 | 0.105 | 0.159 |
| online-G | 0.153 | 0.229 | 0.229 | 0.237 | 0.242 | 0.161 | 0.190 | 0.119 | 0.195 |

Table 8: Entropy values for the English-Czech test suite (consistency features).

→ *the book that I* ***will*** *read.*

## 3 Results

In Tables 2–5, we summarize the WMT18 submissions of the four language directions in terms of BLEU scores[10] and human evaluation scores on the official test set (Bojar et al., 2018).[11]

In the following, we present the results of all our tests across languages in an as uniform way as possible. Bolding in the tables means simply the best result in that category. We do not use any significance tests here. All tables are sorted according to the human evaluation scores.

### 3.1 English–Czech

Results for the paradigm contrast features in English–Czech are shown in Table 6. Not taking into account *online* systems whose architectures are unknown, the table shows a contrast between a Recurrent Neural Network model (*uedin*) and a Transformer model (*CUNI-Transformer*). The former obtains slightly higher accuracies than the latter. This is especially obvious in verb tasks (past and conditional), as well as for noun number. This might suggest that Transformer models have more difficulty in conveying a morphological feature from source to target.[12]

However, we observe no such difference for agreement features (Table 7), where *uedin* obtains an average accuracy of 87.3 and *CUNI-Transformer* obtains 87.9. The latter is slightly better for coordinated verbs and noun phrase inner agreement (see the Adj+Nouns columns), but the former is significantly better in terms of coreference with a relative pronoun (Coref. rel.).

Both systems obtain similar average entropy values in Table 8. These results can be compared to the ones shown in Table 7 of B&Y, although they were computed on another version of the test suite containing different sentences. Whereas the

[10] `http://matrix.statmt.org`

[11] With the exception of Turkish–English, we are not able to compute BLEU scores on the test suite data, as no reference translations are available.

[12] It is however important to note that *not* preserving past of conditional form of the verb needs not lead to a lower translation quality in general because in many situations, less precise wording does not really affect the overall meaning. The reader may subconsciously correct smaller discrepancies among sentences while enjoying the more fluent or more common wording.

|  | Verbs |  |  |  | Pronouns | Nouns |  | Adjectives |  | Average |
|---|---|---|---|---|---|---|---|---|---|---|
| System | Past | Future | Cond. | Neg. | Plur. | Compd. | Nbr. | Compar. | Superl. |  |
| online-Z | 85.0 | 42.2 | 79.2 | 95.8 | 97.7 | 63.1 | 58.2 | 87.6 | 93.9 | 78.1 |
| online-B | 91.3 | 86.8 | 92.3 | 98.4 | 99.2 | 63.7 | 67.3 | 92.8 | 98.7 | 87.8 |
| Microsoft-Marian | 90.4 | 71.3 | 97.6 | **99.4** | 98.6 | 63.6 | 65.2 | 94.9 | 99.6 | 86.7 |
| MMT-production-system | 92.3 | 79.7 | 86.4 | 98.4 | 97.3 | **67.2** | 63.1 | 93.1 | 98.9 | 86.3 |
| UCAM | 94.7 | 84.6 | 98.0 | 99.2 | 99.0 | 64.0 | 68.0 | **97.5** | **100.0** | **89.5** |
| NTT | 93.9 | 89.2 | 97.2 | 99.2 | 99.6 | 61.2 | 68.5 | 96.5 | **100.0** | **89.5** |
| KIT | 89.6 | 74.4 | 96.6 | 98.8 | 98.8 | 61.9 | 64.9 | 93.4 | 99.6 | 86.4 |
| online-Y | 91.5 | 81.4 | 91.3 | 98.8 | 98.8 | 66.1 | 67.3 | 94.0 | 99.1 | 87.6 |
| JHU | 92.6 | **90.4** | 94.6 | 97.4 | 99.6 | **67.2** | 69.0 | 93.6 | 98.9 | 89.3 |
| uedin | 93.1 | 79.4 | 97.4 | **99.4** | 97.2 | 66.4 | 65.4 | 94.4 | 99.1 | 88.0 |
| LMU-nmt | 93.6 | 80.2 | 98.4 | **99.4** | 97.3 | 66.8 | 70.9 | 94.9 | 99.8 | 89.0 |
| online-A | 93.5 | 87.5 | 95.4 | 99.2 | 99.4 | 62.6 | **71.9** | 95.5 | 99.1 | 89.3 |
| online-F | **98.7** | 2.1 | 98.4 | **99.4** | **100.0** | 63.4 | 70.5 | 95.1 | 99.3 | 80.8 |
| online-G | 90.2 | 52.8 | 92.8 | 98.8 | 95.8 | 54.2 | 63.1 | 90.7 | 97.5 | 81.8 |
| RWTH-UNSUPER | 92.3 | 52.3 | **99.2** | 98.6 | 95.7 | 18.9 | 71.1 | 88.3 | 98.1 | 79.4 |
| LMU-unsup | 74.7 | 45.4 | 97.2 | 88.6 | 93.8 | 58.0 | 67.2 | 84.7 | 99.7 | 78.8 |

Table 9: Accuracy values for the English-German test suite (paradigm contrast features).

|  | Coordinated verbs |  |  | Verb | Adj+Nouns |  | Coref. rel. |  | Cor. per. | Adj. | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| System | Nbr | Pers | Tense | Pos | Gdr | Nbr | Gdr | Nbr | Gdr | Strong |  |
| online-Z | 80.5 | 80.7 | 80.3 | 90.8 | 99.8 | 99.8 | 65.5 | 65.5 | 93.2 | 79.8 | 83.6 |
| online-B | 98.7 | 98.7 | 98.7 | 96.0 | **100.0** | **100.0** | 69.6 | 69.6 | 88.7 | 95.7 | 91.6 |
| Microsoft-Marian | 96.3 | 96.3 | 96.3 | 93.8 | **100.0** | **100.0** | 70.1 | 70.1 | 93.8 | 96.7 | 91.3 |
| MMT-production-system | 92.3 | 92.6 | 92.3 | 94.8 | 99.3 | 99.6 | 68.7 | 68.7 | 94.4 | 94.1 | 89.7 |
| UCAM | 97.4 | 97.4 | 97.4 | 93.8 | **100.0** | **100.0** | 68.4 | 68.4 | 94.8 | 99.1 | 91.6 |
| NTT | 98.9 | 98.9 | 98.4 | 94.8 | **100.0** | **100.0** | 70.0 | 70.0 | 93.3 | 96.8 | 92.1 |
| KIT | 97.0 | 97.0 | 96.7 | 93.4 | **100.0** | **100.0** | 69.6 | 69.6 | 93.4 | 96.1 | 91.3 |
| online-Y | 97.1 | 97.1 | 97.1 | **97.2** | **100.0** | **100.0** | 70.8 | 70.8 | 93.0 | 97.9 | 92.1 |
| JHU | 99.6 | 99.8 | **99.8** | 94.2 | **100.0** | **100.0** | 70.1 | 70.1 | 91.6 | 97.4 | **92.2** |
| uedin | 99.1 | 99.1 | 99.1 | 94.0 | 99.5 | **100.0** | 68.5 | 68.5 | 94.1 | 99.6 | **92.2** |
| LMU-nmt | 98.5 | 99.0 | 99.0 | 96.2 | **100.0** | **100.0** | **72.2** | **72.2** | 84.7 | 96.9 | 91.9 |
| online-A | 99.7 | **100.0** | 99.0 | 88.5 | **100.0** | **100.0** | 70.1 | 70.1 | 85.3 | 98.2 | 91.1 |
| online-F | **99.8** | **100.0** | 99.3 | 92.6 | **100.0** | **100.0** | 68.6 | 68.6 | 90.8 | **100.0** | 92.0 |
| online-G | 99.0 | **100.0** | 99.5 | 56.6 | **100.0** | **100.0** | 65.2 | 65.2 | 78.6 | 92.5 | 85.6 |
| RWTH-UNSUPER | 98.7 | 99.7 | 98.4 | 95.4 | 98.0 | **100.0** | 65.7 | 65.7 | 72.6 | 98.9 | 89.3 |
| LMU-unsup | 97.1 | **100.0** | 96.5 | 88.3 | 99.4 | 99.7 | 45.6 | 45.6 | **95.1** | 99.3 | 86.7 |

Table 10: Accuracy values for the English-German test suite (agreement features).

|  | Nouns | Adjectives |  | Verbs |  |  | Average |
|---|---|---|---|---|---|---|---|
| System | Case | Gender | Number | Number | Person | Tense |  |
| online-Z | 0.038 | 0.034 | 0.030 | 0.069 | 0.055 | 0.091 | 0.053 |
| online-B | 0.015 | 0.010 | 0.008 | 0.025 | 0.013 | 0.055 | 0.021 |
| Microsoft-Marian | 0.014 | 0.006 | 0.004 | 0.022 | 0.015 | 0.051 | 0.019 |
| MMT-production-system | 0.015 | 0.017 | 0.015 | 0.031 | 0.020 | 0.071 | 0.028 |
| UCAM | 0.015 | 0.007 | 0.005 | 0.014 | 0.006 | 0.048 | 0.016 |
| NTT | 0.015 | **0.001** | **0.000** | 0.019 | 0.012 | 0.049 | 0.016 |
| KIT | 0.017 | 0.009 | 0.008 | 0.024 | 0.017 | 0.059 | 0.022 |
| online-Y | 0.019 | 0.013 | 0.011 | 0.033 | 0.019 | 0.073 | 0.028 |
| JHU | 0.009 | 0.007 | 0.006 | 0.027 | 0.013 | 0.063 | 0.021 |
| uedin | 0.011 | 0.005 | 0.003 | 0.024 | 0.017 | 0.051 | 0.019 |
| LMU-nmt | 0.020 | 0.003 | 0.003 | 0.023 | 0.008 | 0.067 | 0.021 |
| online-A | 0.015 | 0.003 | 0.001 | 0.037 | 0.011 | 0.070 | 0.023 |
| online-F | **0.005** | 0.002 | 0.001 | **0.011** | **0.004** | **0.034** | **0.010** |
| online-G | 0.030 | 0.006 | 0.001 | 0.068 | 0.014 | 0.087 | 0.034 |
| RWTH-UNSUPER | 0.031 | 0.015 | 0.010 | 0.060 | 0.009 | 0.115 | 0.040 |
| LMU-unsup | 0.019 | 0.015 | **0.000** | 0.098 | 0.015 | 0.137 | 0.047 |

Table 11: Entropy values for the English-German test suite (consistency features).

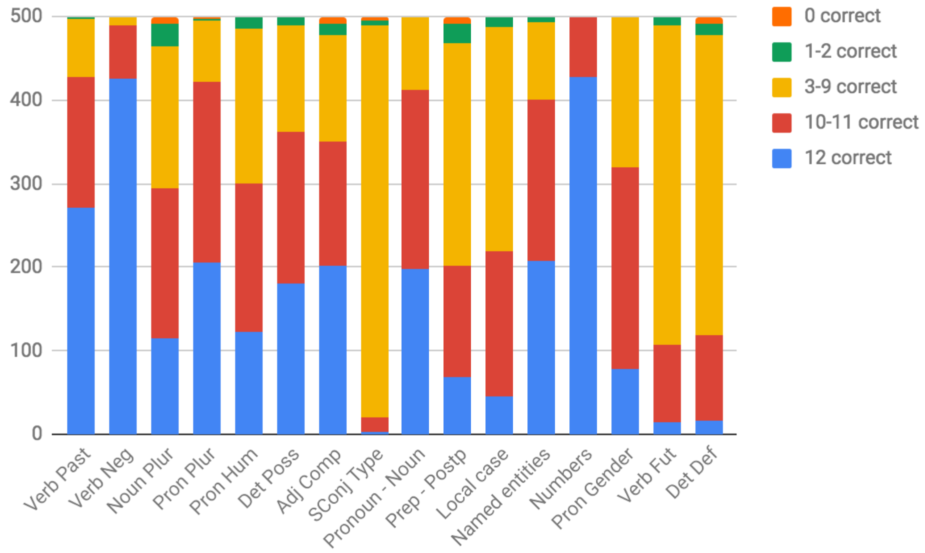


Figure 1: Distribution of correct labels across examples for English–Finnish. *n correct* represents the number of examples (out of the total 500 per contrast) for which $n$ systems (out of a total of 12) were able to generate the contrast correctly.

best system listed there (LIMSI FNMT) obtained an average entropy of 0.168, the WMT 2018 systems *uedin* and *CUNI-Transformer* turn out to be significantly lower (0.125 and 0.131, respectively).

### 3.2 English–German

Results for the paradigm contrast features in English–German are shown in Table 9. It is clear from the table that certain tasks are now too easy for the current state-of-the-art: verb negation, pronoun plural and superlative are very close to a perfect accuracy across nearly all systems. The hardest task seems to be the one involving compound generation (Nouns Compd. in Table 9), where accuracies range from 18.9 to 66.4. Verb future tense also causes considerable difficulties to several systems, including the top-scoring online-Z. As with English–Czech, we see that the systems best ranked according to manual evaluation (closer to the top of the list) do not necessarily score well in this detailed evaluation and vice versa. One example is the anonymous *Online-Z* system, which is rather bad at preserving verb attributes, noun number or comparative adjectives.

Table 10 shows even more clearly how easy certain tasks are. Indeed, noun phrase internal agreement (gender and number) seems to be perfectly modeled by every system (accuracies range from 98.0 to 100, see the columns Adj+Nouns). Coordinated verbs and strong/weak adjectives seem rather easy as well, with all accuracies over 90%. Coreference with relative pronouns (Coref. rel.) seems to be the most difficult task. Note that we observe exactly the same results for gender and number: this is due to the fact that the SMOR analysis of relative pronouns is highly ambiguous. E.g. the pronoun *die* is both singular and plural, and has no specific gender in plural form, therefore it may agree with any noun. Strictly all the errors for this task are due to the fact that we could not find the right noun or pronoun in the sentence, which leads to no difference between gender and number. Hence the task does not measure agreement as much as the ability of a system to output a relative pronoun.

Consistency tasks are shown in Table 11. Strikingly, the *online-Z* system, ranked best on human judgement, shows the worst entropy score. Overall, the consistency task figures do not seem to correlate well with general translation quality measures. Compared to to the Czech values in Table 8, we notice that the German average entropy values are quite low. This could be explained by the fact that Czech has a richer nominal, adjectival and verbal morphology than German. For instance, whereas German has four cases, Czech has seven, which impacts the entropy values computed for this task.

|  | Verbs |  | Nouns | Pronouns |  | Det | Adj | SConj |  |
|---|---|---|---|---|---|---|---|---|---|
| System | Past | Neg | Plur | Plur | Hum | Poss | Compar | Type | Average |
| NICT | 94.4 | 98.6 | 79.2 | 94.6 | 90.4 | 88.4 | 88.0 | **96.2** | **91.2** |
| HY-NMT | 93.8 | **99.0** | 74.8 | 82.6 | 67.4 | 83.6 | 78.6 | 96.0 | 84.5 |
| uedin | 94.0 | 98.8 | 75.0 | 93.6 | 82.6 | 85.0 | 87.8 | 90.2 | 88.4 |
| Aalto | 93.4 | 98.8 | 72.0 | 88.4 | 77.4 | 90.8 | 81.6 | 87.6 | 86.3 |
| HY-NMT2step | **95.2** | 99.0 | 69.8 | 91.4 | 83.8 | **94.0** | 81.6 | 87.4 | 87.8 |
| talp-upc | 91.0 | 98.4 | 72.2 | 94.0 | 80.2 | 83.0 | 79.8 | 84.4 | 85.4 |
| CUNI-Kocmi | 89.0 | 98.0 | 73.8 | 91.4 | 80.4 | 86.2 | 78.6 | 76.6 | 84.3 |
| online-B | 92.0 | 98.6 | 76.4 | 91.0 | 78.0 | 77.0 | 82.2 | 84.8 | 85.0 |
| online-A | 87.6 | **99.0** | 78.6 | 94.2 | 82.0 | 84.6 | 86.0 | 23.4 | 79.4 |
| online-G | 82.8 | 92.6 | 76.8 | 86.8 | 66.2 | 83.0 | **88.2** | 3.2 | 72.5 |
| HY-SMT | 79.0 | 96.2 | 53.2 | 62.8 | 59.4 | 80.6 | 68.8 | 6.8 | 63.4 |
| HY-AH | 93.4 | 98.2 | **88.8** | **99.0** | **94.8** | 76.0 | 87.2 | 1.0 | 79.8 |

Table 12: Accuracy values for the English–Finnish test suite (paradigm completion features).

| System | Adj+ Noun | Prep / Postp | Local case | Average |
|---|---|---|---|---|
| NICT | **96.2** | **88.2** | 81.2 | **88.5** |
| HY-NMT | 87.8 | 81.8 | 68.6 | 79.4 |
| uedin | 92.0 | 83.0 | 80.4 | 85.1 |
| Aalto | 93.2 | 81.4 | 69.8 | 81.5 |
| HY-NMT2step | 90.0 | 86.8 | 70.4 | 82.4 |
| talp-upc | 91.8 | 70.4 | 77.4 | 79.9 |
| CUNI-Kocmi | 91.4 | 63.8 | 71.2 | 75.5 |
| online-B | 90.2 | 72.8 | 66.0 | 76.3 |
| online-A | 81.2 | 41.4 | 78.4 | 67.0 |
| online-G | 84.6 | 33.8 | 80.2 | 66.2 |
| HY-SMT | 78.6 | 48.0 | 41.4 | 56.0 |
| HY-AH | 89.8 | 74.0 | **81.8** | 81.9 |

| System | Named entities | Numbers | Average |
|---|---|---|---|
| NICT | 90.4 | 99.4 | 94.9 |
| HY-NMT | 91.6 | 98.4 | 95.0 |
| uedin | 92.4 | 99.8 | 96.1 |
| Aalto | 82.4 | 96.0 | 89.2 |
| HY-NMT2step | 81.8 | 97.0 | 89.4 |
| talp-upc | 79.8 | 98.8 | 89.3 |
| CUNI-Kocmi | 86.6 | 99.8 | 93.2 |
| online-B | **94.8** | 99.0 | **96.9** |
| online-A | 90.2 | 99.8 | 95.0 |
| online-G | 86.2 | **100.0** | 93.1 |
| HY-SMT | 81.6 | 93.8 | 87.7 |
| HY-AH | 85.0 | 99.8 | 92.4 |

Table 13: Accuracy values for the English–Finnish test suite (left: agreement features, right: rare word features).

| System | Verbs Fut | Pronouns Gender | Det Def | Average |
|---|---|---|---|---|
| NICT | 68.4 | 87.0 | 70.6 | 75.3 |
| HY-NMT | 65.0 | 84.2 | 58.8 | 69.3 |
| uedin | **73.0** | 84.6 | 65.4 | 74.3 |
| Aalto | 71.2 | 74.8 | 63.6 | 69.9 |
| HY-NMT2step | 64.4 | 75.4 | 57.2 | 65.7 |
| talp-upc | 61.0 | 75.0 | 53.2 | 63.1 |
| CUNI-Kocmi | 54.0 | 65.6 | 48.8 | 56.1 |
| online-B | 68.8 | 88.6 | 55.2 | 70.9 |
| online-A | 59.6 | 84.8 | 70.2 | 71.5 |
| online-G | 62.2 | 91.0 | 73.6 | 75.6 |
| HY-SMT | 33.8 | 79.6 | 42.2 | 51.9 |
| HY-AH | 71.4 | **95.0** | **89.0** | **85.1** |

Table 14: Accuracy values for the English–Finnish test suite (stability features).

|  | **Verbs** |  |  |  | **Obj. Part.** | **Subj. Part.** | **Average** |
|---|---|---|---|---|---|---|---|
| System | Person | Future | Past | Neg. | Future | Future |  |
| online-G | 60.0 | 67.3 | 75.5 | 68.3 | 41.0 | 21.8 | 55.65 |
| online-A | **71.3** | **72.3** | **77.3** | **72.0** | **49.5** | **30.5** | **62.15** |
| online-B | 46.8 | 66.8 | 76.3 | 66.5 | 40.3 | 26.8 | 53.92 |
| uedin | 53.5 | 65.0 | 66.5 | 64.5 | 39.0 | 17.0 | 50.92 |
| NICT | 57.8 | 69.0 | 73.3 | 67.8 | 45.5 | 22.3 | 55.95 |

Table 15: Accuracy values for the Turkish–English test suite.

### 3.3 English–Finnish

As a general overview of the English–Finnish features and their difficulty, Figure 1 shows the distribution of correct labels across examples and features. It can be seen that some features (e.g., verb negation or numbers) pose very few problems to current MT systems, whereas others (e.g. subordinate clause type, see SConj Type in the figure) are much more difficult. In contrast to German, the pronoun plural feature seems to be harder for Finnish systems. In particular, the *0 Correct* and *1-2 Correct* categories may indicate potential problems in the example generation or scoring process.

We performed a manual analysis of a small sample of contrast pairs (20-30 examples per feature) regarding the grammaticality of the automatically generated sentences and the recall of the automatic evaluation script. For the features *Noun Plur, Pron Hum, Det Poss, Adj Compar Adj* and *Local case*, more than 20% of the annotated examples showed either problems in the source sentence (incomplete sentences due to splitting errors, ungrammatical or meaningless sentences due to tagging errors, complete meaning changes, etc.), or problems with the evaluation method. Errors of the first class however may not necessarily affect the results of the test suite, as most systems handle incomplete or meaningless sentences rather well. Still, the results of the mentioned features may not be as reliable as those of the remaining ones.

The paradigm completion features (Table 12) show a clear advantage for those two systems that explicitly model target morphology, *HY-NMT2step* and *HY-AH*. On average, these two systems are however outperformed by the *NICT* system, confirming its first rank in the manual evaluation. Most other NMT systems yield comparable accuracies, but it is striking to see that *uedin* repeatedly ranks higher than *HY-NMT* despite its lower BLEU and manual evaluation scores. The only submitted SMT system, *HY-SMT*, clearly underperforms in almost all features. The rule-based *HY-AH* system shows good overall performance, but is penalized by its complete failure on the subordinate clause type task, probably due to some missing or defective rules. We manually checked some examples of the subordinate clause feature, as several systems completely failed on it, and are able to confirm that these systems were indeed unable to correctly generate indirect questions.[13]

The agreement features (left half of Table 13) show a somewhat different picture, with the NICT system clearly leading the board, suggesting that good data selection strategies may be more important for these types of features than explicit modeling of morphology. Still, the *HY-NMT2step* and *HY-AH* yield better scores than their official rankings would suggest.

The rare word features (right half of Table 13) surprise by the exceptional performance of the online systems. It is likely that these systems contain some type of copy mechanism to handle out-of-vocabulary words, whereas such mechanisms are typically not included in research systems. The participating NMT systems use three different subword splitting algorithms: *Aalto* uses Morfessor, *talp-upc* and *CUNI-Kocmi* use wordpieces as implemented in Tensor2Tensor, and *NICT, HY-NMT* and *uedin* use byte-pair encoding. The results suggest that byte-pair encoding performs better than its competitors, but a more careful analysis would be required to confirm this hypothesis. The best performance in rare word features is achieved by online systems B and G, but without knowledge of their internals, we cannot link this performance to training data or dedicated components.

Although a large-scale manual evaluation of the sentence pairs was not within the scope of this paper, a number of English-to-Finnish sentence pairs were extracted for a manual "sanity check". In particular, we focused on cases where only the rule-based system output was evaluated as correct, in order to identify potential false positives/negatives caused by the equally rule-based scoring procedure. One observed weakness of the scoring procedure is that it favors more literal (word-for-word) renderings of the source. This tendency produces false negatives in the cases where the NMT output contained a less literal translation, which may however be both fluent and adequate. False positives can also be observed in some cases where the literal translation in the RBMT output, marked correct, is in fact not a correct translation of the source. These often involved idiomatic expressions (such as *This brings us to X*), which occasionally occur in the sentence pairs even though idioms had been excluded to the ex-

[13] Most failing translations used one of the following constructions: *Hän kysyi, jos se ei tapahtuisi Kaliforniassa. / Hän pyyti jos se ei tapahtuisi Kaliforniassa.* 'She asked if it would not happen in California.'

tent possible.

The stability features (Table 14) show lower figures on average. As could be expected, the rule-based system is the most stable one, as it explicitly encodes the mappings between English and Finnish morphological categories. The online systems again performed quite well on these features. Again, the SMT system is worse than the NMT systems, something that was not necessarily expected, as SMT systems tend to produce more literal translations than NMT systems. Similarly to the German consistency features, the Finnish stability features do not seem to correlate strongly with the human judgement scores. In particular, the poor scores of *CUNI-Kocmi* are surprising and not expected from the other features.

As noted above, stability is not necessarily always a reflection of overall quality, and it may not always be most adequate to produce identical translations for sentence pairs differing in only one feature (verb tense, pronoun gender, definiteness). An interesting example of this was observed in the case of indefinite and definite determiners. As Finnish lacks determiners, translations for sentences involving the definiteness contrast were expected to be identical. This was generally the case for the RBMT system, but NMT systems were observed to produce sentences with word order changes that are used in Finnish to indicate distinctions corresponding to the English definite/indefinite articles. The sample extracted for this manual check is insufficient to determine whether these word order differences can be considered something the NMT system has learned from the corpus or simply random variation, but the observation that they occur is interesting. Certainly, NMT systems do have the capacity to learn to express sentence information structure but it is not yet clear if it is sufficiently exemplified in the training data.

An overall point should also be made that the sentence pair evaluation only compares the specific feature being evaluated, or compares whether the sentences are identical in the case of the stability features. The overall correctness, adequacy or fluency is not evaluated, and sentences evaluated as correct for a specific feature may – and indeed often do – contain other errors or problems.

### 3.4 Turkish–English

Finally, we present our evaluation results for Turkish–English in Table 15.[14]

We can observe that none of the systems perform particularly well on either of the participle contrast pairs. Interestingly, performance is worse on the more frequent subject participles. There is also a stark difference in performance across different systems in subject participles, with *Online-A*'s accuracy (30.5%) being almost twice that of *uedin* (17.0%).

Again, the overall translation performance is not quite in line with the performance on our test suite.

## 4 Conclusions

The contrastive evaluation of morphological competence, as introduced by B&Y, has proved to be easy to adapt to additional language pairs and linguistic features. The data collected from the systems participating in WMT18 allows for fine-grained analysis of the impact of system architectures, training parameters and data on the various aspects of morphological competence. In general, the systems that perform well on global quality evaluation also show good morphological competence, but a few striking differences have been found. First, rule-based systems such as *HY-AH* for English–Finnish tend to obtain much higher morphology scores than expected from their overall quality. This is not surprising, as rule-based systems usually contain an explicit morphological generation component, but it requires more research on the factors that influence the correlation between morphological tests and overall translation quality. Second, we found that features focusing on consistency and stability (i.e., those presented in Tables 8, 11 and 14) correlate poorly with human judgement. This suggests that the robustness of current MT system has almost no relation to their quality.

## Acknowledgments

This research has been supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 780069 and by the grant 18-24210S of the Czech Science Foundation.

[14] Unfortunately, we did not obtain the output of the *Alibaba-Ensemble* system in time for evaluation.

## References

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Franck Burlot and François Yvon. 2017. Evaluating the morphological competence of machine translation systems. In *Proceedings of the Second Conference on Machine Translation*, pages 43–55, Copenhagen, Denmark. Association for Computational Linguistics.

Franck Burlot and François Yvon. 2018. Evaluation morphologique pour la traduction automatique: adaptation au français. In *Conférence sur le Traitement Automatique des Langues Naturelles*, TALN, Rennes, France. ATALA.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan. Association for Computational Linguistics.

Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Krister Lindén, Erik Axelson, Sam Hardwick, Tommi A. Pirinen, and Miikka Silfverberg. 2011. HFST – framework for compiling and applying morphologies. In *Proceedings of the International Workshop on Systems and Frameworks for Computational Morphology*, pages 67–85. Springer, Berlin, Heidelberg.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

George A. Miller. 1995. WordNet: A lexical database for English. *Commun. ACM*, 38(11):39–41.

Tommi A. Pirinen. 2015. Omorfi —free and open source morphological lexical database for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 313–315, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition, and inflection. In *Proceedings of the IVth International Conference on Language Resources and Evaluation (LREC 2004*, pages 1263–1266.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajič. 2014. Open-source tools for morphology, lemmatization, POS tagging and named entity recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

# Testsuite on Czech–English Grammatical Contrasts

**Silvie Cinková** **Ondřej Bojar**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
<surname>@ufal.mff.cuni.cz

## Abstract

We present a pilot study of machine translation of selected grammatical contrasts between Czech and English in WMT18 News Translation Task. For each phenomenon, we run a dedicated test which checks if the candidate translation expresses the phenomenon as expected or not. The proposed type of analysis is not an evaluation in the strict sense because the phenomenon can be correctly translated in various ways and we anticipate only one. What is nevertheless interesting are the differences between various MT systems and the single reference translation in their general tendency in handling the given phenomenon.

## 1 Introduction

English and Czech are typologically different languages. It goes without saying that some structural phenomena of either lack a direct structural equivalent in the other; for instance, Czech has not grammaticalized noun definiteness, while it boasts a complex system of verb aspect, which is absent in English. Such *1:n* correspondences can pose translation problems in human as well as in machine translation. Intuitively, a translation system that has mastered these *1:n* phenomena ought to be more successful than one that has not. Therefore we investigate whether there is a positive correlation between mastering some of these problematic phenomena and the performance of an En-Cs MT system.

## 2 Selected Linguistic Phenomena

Based on our experience as Czech learners of English, translators and developers/evaluators of MT systems, we have selected the following phenomena for EN-CS translation evaluation: **English gerundial clause** and English verb control with **controlled infinitive**.

The data comes from a manually-parsed, word-aligned parallel treebank of English news texts and their human Czech translations (see Section 3).

### 2.1 English gerundial clause (and other *ing*-forms)

Modern Czech has no counterpart of the English gerund. Older Czech (i.e. until approximately 1950), used to have a verb form called present **transgressive**, which would be very handy to translate many cases of English gerundial clauses, but this form is perceived as archaic and hardly ever used. Modern Czech has the following options to render the English gerund:

1. finite clause with a choice of subordinators or conjunctions;

2. non-finite clause (infinitive clause, nominalization, or adjective/present participle).

In this study we tested whether the Czech equivalent in the reference vs. automatic translation was a finite clause or anything else.

#### 2.1.1 Czech finite clause as equivalent to English gerundial clause

Czech is more sensitive to convoluted expressions than English. Therefore non-finite clauses are usually most smoothly translated with finite clauses. To keep the Czech text coherent, though, human translators usually link the gerundial clause to the main clause with an explicit discourse connective – either a conjunction or a subordinator, based on their knowledge of context and their world knowledge. This may pose a challenge for MT systems. The most typical discourse connectives used to translate gerundial clauses would be *-li* (a clitic *if* or *whether*), *což* (*which* referring to a predicate), *protože* (*because*), *když* (*when*), *že* (*that* as subordinator), *jak* (approximately *as* expressing

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 561–569
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64061

an event parallel to the main-clause event), and *a* (*and*). Example:

(1) When they arrived at the door, all were afraid to go in, fearing that they would be out of place.
Ale když přišli ke dveřím, všichni se báli vstoupit, protože se báli, že budou působit trapně.
(But when they arrived at the door, all were afraid to go in, because they feared that they would be out of place.[1] )

(2) He said he was surprised by the EC's reaction, calling it "vehement, even frenetic."
Řekl, že byl překvapen reakcí ES, a nazval ji "prudkou, ba i bouřlivou".
(He said he was surprised by the EC's reaction, and he called it "vehement, even frenetic".)

### 2.1.2 Czech infinitive as equivalent to English gerundial clause

Infinitive clause occurs in our sample to translate gerundial clauses in the subject position and in control in some verbs. Example:

(3) Avoiding failure is easy.
Vyhnout se neúspěchu je snadné.
(To avoid failure is easy.)

(4) So far no one has suggested putting the comptroller back on the board.
Zatím nikdo nenavrhl znovu dosadit do Rady také kontrolora.
(So far no one has suggested to put the comptroller back on the board.)

### 2.1.3 Nominalizations as equivalents to gerundial clause

The choice between deverbal noun and event noun is lexically motivated. A deverbal noun is a noun derived from a verb stem by suffixes *-ní, -tí*; e.g. *stát* v. – *stání* n., *proklít* v. – *prokletí* n. This is an almost universal derivational mechanism, but it is stylistically associated with officialese and easily overused.

An event noun is a noun with either no derivative relation to any semantically close verb stem (*restaurace*, n. – NULL v.[2]) or a less productive derivation relation to a verb stem; e.g. *podpořit* v. – *podpora* n., *letět* v. – *let* n.). Also these nominalizations are to be used sparingly to preserve readability.

Example:

(5) Consider adopting your spouse's name.
Zvažte přijetí příjmení svého partnera.
(Consider the adopting of your spouse's name.)

(6) The Canadian wound up writing a check.
Kanaďan ukončil vysvětlování vypsáním šeku.
(The Canadian wound up with the writing of a check. )

(7) Fear of AIDS hinders hiring at few hospitals.
Strach z AIDS komplikuje nábor v několika nemocnicích.
(Fear of AIDS hinders recruitment at few hospitals.)

### 2.1.4 Present participle as equivalent to gerundial clause

The Czech present participle is derived from a verb but behaves like a regular adjective, including inflection; e.g. *spát* v. – *spící* adj.

As an equivalent to the English gerundial clause it requires a syntactic transformation of the source clause, approximately as though the original clause contained a participial clause instead of the gerund. Square brackets in the following example show the syntactic dependencies in English *imagined by the translator* and the corresponding structure in Czech. The main predicate is typeset in bold. Example:

(8) [[Mr. Fukuyama, [peering]] through binoculars at the end of history, **said**] ... [[Francis Fukuyama [nakukující]] skrz brýle na konci historie, ... **uvádí**], že ...

(9) [Other steelmakers **envision** steel [roofs [covering]] suburbia.] [Další výrobci oceli si **představují** ocelové [střechy [pokrývající]] předměstí.]

[1] To make the structure of the target *Czech* reference sentence more accessible to non-Czech speakers, we enhance this paper with their literal English translations. We enclose these—naturally awkward—sentences in parentheses.

[2] The verb *restaurovat* means *restore*, whereas the noun *restaurace* means *restaurant*.

### 2.2 English infinitive clause

The English infinitive clause has many functions; e.g. verb control or a convoluted subordinate clause. Infinitive as controlled verb in verb control is present in both languages, but the many other uses of the English infinitive clause have different structure equivalents in Czech—mostly different types of finite subordinate clauses. A correct parsing would possibly make it easier for an MT system to select a plausible Czech equivalent structure, but the parser was not able to reliably identify the correct syntactic governing node of an infinitive clause in our data sample.

Since we could not rely on the parser to tell infinitive clause as an argument from an adjunct, we did not limit our search to arguments. Our sample contains the following Czech structural equivalents to English infinitive clauses:

1. **infinitive** or **noun phrase**;
2. **finite clause**.

#### 2.2.1 Infinitive as controlled verb

A proportion of verb control cases have a *1:1* translation to Czech.

Example:

(10) Comair said it paid cash but *declined* **to disclose** the price.
Společnost Comair uvedla, že zaplatila hotově, avšak *odmítla* **uvést** cenu.

However, many English controlling verbs have a Czech equivalent verb that cannot act as a controlling verb. To avoid a verbose paraphrase with an expletive pronoun and a subordinate content clause, Czech can resort to a nominalization (deverbal noun or event noun; see Section 2.1.3):

Example:

(11) Mr. Friend says he agreed to strike Mr. Alexander above the belt.
Pan Friend říká, že souhlasil s udeřením pana Alexandera nad opaskem.
(Mr. Friend says he agreed with a striking of Mr. Alexander above the belt.)

The verbose translation would say:

(12) Pan Friend říká, že souhlasil s tím, že udeří pana Alexandera nad opaskem.
(Mr. Friend says he agreed with it that he would strike Mr. Alexander above the belt.)

#### 2.2.2 Finite clause as equivalent to English infinitive clause

English has an infinitive structure that resembles a consecutive clause but involves a semantic shift towards temporal sequence of two events. This structure exists in Czech, too, but it is not common. A more natural translation would use a coordination of finite clauses. Example:

(13) The stock gained $2.75 Thursday to close at a then-52 week high.
Cenný papír ve čtvrtek navýšil o 2.75 dolaru a uzavíral na vrcholu tehdejších 52 týdnů.
(The stock gained $2.75 Thursday and was closing at a then-52 week high.)

Purpose and consecutive clauses, as well as content clauses, are typically finite in Czech, using a range of subordinators (cf. Section 2.1.4).

Examples:

(14) It also redesigned Oil of Olay's packaging, stamping the traditional pink boxes with gold lines **to create** a more opulent look.
Společnost rovněž změnila obal krému, na tradiční růžová políčka přidala zlaté linky, **čímž vytvořila** lukrativnější vzhled.
(It also redesigned Oil of Olay's packaging, stamping the traditional pink boxes with gold lines, by which it created a more opulent look.)

(15) At least three other factors have encouraged the IMF to insist on increased capital.
Nejméně tři další faktory přiměly MMF k tomu, aby na zvýšení kapitálu trval.
(At least three other factors have encouraged the IMF to that it should insist on increased capital.)

## 3 Data Set

Our sentences come from the Prague Czech-English Dependency Treebank 2.0 (PCEDT 2.0) (Hajič et al., 2012). PCEDT 2.0 is a multi-layered parallel treebank with automatic word alignment, manually built upon the Penn Treebank (Marcus et al., 1994) and its translation into Czech. It has two syntactic layers of rooted dependency trees with labeled edges: the analytical (*a*-) layer with surface syntax and the tectogrammatical (*t*-) layer with deep syntax.

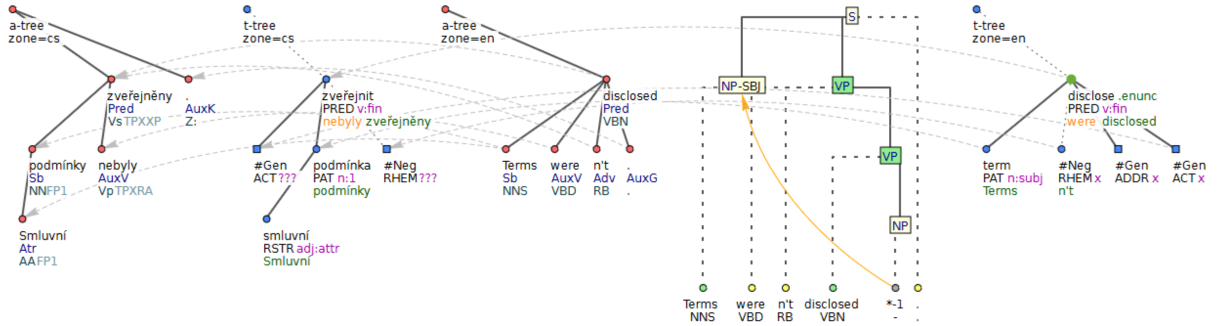


Figure 1: A sentence representation in PCEDT 2.0. The English part also contains the original PennTreebank.

In the a-layer, each word token is represented by one node. The inner structure of each node contains the word form, lemma, POS-tag, dependency label (*afun*), and reference to the governing node. The t-layer represents the linguistic meaning of each sentence by a tree that somewhat abstracts from details of morphology and surface syntax, but remains, by and large, a syntactic dependency tree. Each node contains references to the a-layer corresponding a-layer node(s), along with a whole range of other attribute values. Different reference types to content and auxiliary words, respectively. Apart from that, the t-layer provides semantic role labeling (functors), as well as coreference and ellipsis resolution.

Figure 1 illustrates the data structure of PCEDT 2.0 including the alignment links pointing from English to Czech.

We have automatically selected 3782 sentences, using the the PMLTQ search query engine (Štěpánek and Pajas, 2010), using the Czech counterpart of the corpus as reference translation. All the pre-selected sentences were included in inputs of MT systems participating in the WMT18 News Translation Task. In addition to the "primary" systems CUNI Transformer, UEDIN and the online systems, we also added three baseline (contrastive) systems: CUNI Chimera, CUNI Chimera noDepfix and CUNI Moses.

CUNI Transformer is a carefully trained system (Popel and Bojar, 2018) based on the Transformer architecture (Vaswani et al., 2017) and thus without recurrent connections.

UEDIN is an ensemble of deep RNN systems translating left-to-right and reranked by a deep right-to-left RNN model.

CUNI Moses serves as the ultimate baseline. It is phrase-based (Koehn et al., 2007) and trained on a very large parallel corpus and further adapted for the news text.

CUNI Chimera is the hybrid setup that served very well in 2013–2015 (Bojar et al., 2013). A phrase-based backbone is used to combine translations by a transfer-based system TectoMT (Žabokrtský et al., 2008), by Nematus (Sennrich et al., 2017) and by Neural Monkey (Helcl et al., 2018) with phrase pairs from the large parallel corpus. The final step of Chimera was the application of a dependency-based automatic error correction tool Depfix (Rosa et al., 2012). In this paper we report the performance of both the full CUNI Chimera and a version without a the depfix post-correction, labelled CUNI Chimera noDepFix.

Since our sentences originally come from the WSJ section of the Penn Treebank, they belong to the domain of the translation task.

## 4 Evaluation

For each phenomenon we implemented a small test relying on an automatic analysis of the source English to the surface syntactic tree (a-layer, in the terminology of PCEDT), an automatic analysis of the Czech translation to surface (a-layer) along with a deep (t-layer) syntactic tree, and on automatic word alignments between the English a-layer and Czech a-layer and t-layer. We aligned directly English to each of the Czech layers; a more rigorous approach would have been aligning only the a-layers and follow the links between a-layer and t-layer on the Czech side, but since all our annotations are automatic, we do not expect much difference in these approaches due to random errors in all processing steps. The annotation was provided by the pipeline used in the creation of corpus CzEng (Bojar et al., 2016)[3]

[3] http://ufal.mff.cuni.cz/czeng

as implemented in the Treex toolkit (Popel and Žabokrtský, 2010). For the alignment, we relied on an intersection of GIZA++ (Och and Ney, 2000) alignments.

The test searched for the keyword related to the phenomenon (e.g. the controlled English verb), followed the word-alignment links to the tested some morphological or syntactic properties of the corresponding Czech word or node in t-layer analysis. The result of the test was "Good" if the Czech expression was the best possible translation, "Bad" otherwise, and "Unknown" if the target word or node was not found, e.g. due to errors in word alignment.

It is important to note that "Bad" does not always mean an inacceptable translation. It merely means that the translation is not the most straighforward one.

Table 1 below presents the detailed results of these tests.

While the manual evaluation of WMT18 systems is not yet available, we can assume that it will match the automatic evaluation available at http://matrix.statmt.org/matrix/systems_list/1883 and reproduced here in Table 2. One caveat to keep in mind is that this evaluation is based on a different set of sentences than we use in our testsuite.

Disregarding the "Unknowns", we plot the results in Figure 2 and Figure 3 by systems and by phenomena, respectively.

## 5 Discussion

One observation is that the reference generally adopts the most typical translation in all the phenomena. (A small exception is the performance of UEDIN in EN-gerund-CS-finclause on the refined set; not confirmed on the larger set though.) At the same time, the reference does not always match our expectation. The most divergent phenomenon is EN-gerund-CS-finclause where the reference uses the expected finite clause only in $56/(56 + 25.3) = 68.9\%$ of cases.

In general, the reference seems a little harder to process ("Unk" higher than for MT systems), probably due to a less verbatim translation and thus a less straightforward word alignment.

The order of MT systems does not match their overall automatic performance. CUNI Transformer, the best-performing system overall (and a system that is actually likely to surpass humans this year in sentence-level evaluation) appears in the middle of our list. This suggests that Transformer outputs may be "more creative", departing more from the reference. UEDIN, on the other hand, seems to be very close to the reference in the studied phenomena. Finally, phrase-based Moses has been clearly surpassed in all evaluations. As a next step, we plan to obtain and compare manual evaluations of individual sentences . The annotators will rate the automatic and reference translations alike, without knowing which is which. For each system, we will compare the correlation between the quality rating and the agreement with the reference translation.

## 6 Conclusion

We have presented a testsuite focused on English–Czech translation of a small set of extremely frequent verb-related phenomena. The testsuite of about 3000 automatically preselected sentences reveals that the two overall top-performing systems UEDIN and CUNI Transformer differ considerably in their handling of the phenomena. Further investigation, esp. in link with the manual annotation which is now running for WMT18, is needed to validate whether the less expected translations for our selected phenomena reflect the assessed translation quality.

The dataset is publicly accessible via the LINDAT-CLARIN repository:

http://hdl.handle.net/11234/1-2856

## Acknowledgments

This work has been supported by the Czech Science Foundation grant 18-24210S and by the Czech Ministry of Education, Youth, and Sports LTC18020. We have used the data, tools, and infrastructure of the LINDAT-CLARIN repository (https://lindat.mff.cuni.cz/en/), Research Infrastructure CZ.02.1.01/0.0/0.0/16_013/0001781.

## References

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Computer Science, pages

231–238, Cham / Heidelberg / New York / Dordrecht / London. Masaryk University, Springer International Publishing.

Ondřej Bojar, Rudolf Rosa, and Aleš Tamchyna. 2013. Chimera – Three Heads for English-to-Czech Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 92–98, Sofia, Bulgaria. Association for Computational Linguistics.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing prague czech-english dependency treebank 2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3153–3160, İstanbul, Turkey. ELRA, European Language Resources Association.

Jindřich Helcl, Jindřich Libovický, Tom Kocmi, Tomáš Musil, Ondřej Cífka, Dušan Variš, and Ondřej Bojar. 2018. Neural monkey: The current state and beyond. In *The 13th Conference of The Association for Machine Translation in the Americas, Vol. 1: MT Researchers' Track*, pages 168–176, Stroudsburg, PA, USA. The Association for Machine Translation in the Americas, The Association for Machine Translation in the Americas.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2000. A Comparison of Alignment Models for Statistical Machine Translation. In *Proceedings of the 17th conference on Computational linguistics*, pages 1086–1090. Association for Computational Linguistics.

Martin Popel and Ondřej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Martin Popel and Zdeněk Žabokrtský. 2010. TectoMT: Modular NLP framework. In *Lecture Notes in Artificial Intelligence, Proceedings of the 7th International Conference on Advances in Natural Language Processing (IceTAL 2010)*, volume 6233 of *Lecture Notes in Computer Science*, pages 293–304, Berlin / Heidelberg. Iceland Centre for Language Technology (ICLT), Springer.

Rudolf Rosa, David Mareček, and Ondřej Dušek. 2012. DEPFIX: A System for Automatic Correction of Czech MT Outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 362–368, Montréal, Canada. Association for Computational Linguistics.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Jan Štěpánek and Petr Pajas. 2010. Querying diverse treebanks in a uniform way. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 1828–1835, Valletta, Malta. European Language Resources Association.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. TectoMT: Highly Modular Hybrid MT System with Tectogrammatics Used as Transfer Layer. In *Proc. of the ACL Workshop on Statistical Machine Translation*, pages 167–170, Columbus, Ohio, USA.

| | | Total | Bad | Good | Unk | Total | Bad | Good | Unk |
|---|---|---|---|---|---|---|---|---|---|
| EN-control-CS-finclause | Reference | 76 | 7.9 | 72.4 | 19.7 | 100 | 8.0 | 71.0 | 21.0 |
| EN-control-CS-finclause | UEDIN | 76 | 34.2 | 53.9 | 11.8 | 100 | 39.0 | 50.0 | 11.0 |
| EN-control-CS-finclause | CUNI Chimera | 76 | ≀ 27.6 | 51.3 | 21.1 | 100 | ≀ 31.0 | 47.0 | 22.0 |
| EN-control-CS-finclause | CUNI Chimera noDepFix | 76 | 27.6 | 51.3 | 21.1 | 100 | 31.0 | 47.0 | 22.0 |
| EN-control-CS-finclause | CUNI Transformer | 76 | ≀ 23.7 | ≀ 56.6 | 19.7 | 100 | ≀ 23.0 | ≀ 60.0 | 17.0 |
| EN-control-CS-finclause | online-B | 76 | 23.7 | ≀ 59.2 | 17.1 | 100 | 27.0 | 53.0 | 20.0 |
| EN-control-CS-finclause | online-A | 76 | 68.4 | 22.4 | 9.2 | 100 | 71.0 | 20.0 | 9.0 |
| EN-control-CS-finclause | online-G | 76 | 71.1 | 18.4 | 10.5 | 100 | ≀ 69.0 | ≀ 21.0 | 10.0 |
| EN-control-CS-finclause | CUNI Moses | 76 | ≀ 67.1 | 17.1 | 15.8 | 100 | ≀ 67.0 | 18.0 | 15.0 |
| EN-control-CS-nofinclause | Reference | 104 | 0.0 | 70.2 | 29.8 | 1819 | 0.6 | 74.2 | 25.2 |
| EN-control-CS-nofinclause | UEDIN | 104 | 20.2 | 64.4 | 15.4 | 1819 | 18.0 | 68.4 | 13.6 |
| EN-control-CS-nofinclause | CUNI Chimera | 104 | 23.1 | 60.6 | 16.3 | 1819 | 20.2 | 62.3 | 17.4 |
| EN-control-CS-nofinclause | CUNI Chimera noDepFix | 104 | 23.1 | 60.6 | 16.3 | 1819 | 20.2 | ≀ 62.4 | 17.4 |
| EN-control-CS-nofinclause | CUNI Transformer | 104 | 26.9 | 54.8 | 18.3 | 1819 | ≀ 18.2 | ≀ 64.9 | 16.9 |
| EN-control-CS-nofinclause | online-B | 104 | 28.8 | 52.9 | 18.3 | 1819 | 18.6 | 61.2 | 20.2 |
| EN-control-CS-nofinclause | online-A | 104 | ≀ 10.6 | ≀ 77.9 | 11.5 | 1819 | ≀ 8.7 | ≀ 79.2 | 12.0 |
| EN-control-CS-nofinclause | online-G | 104 | 11.5 | 76.0 | 12.5 | 1819 | ≀ 7.3 | ≀ 81.3 | 11.4 |
| EN-control-CS-nofinclause | CUNI Moses | 104 | ≀ 5.8 | ≀ 80.8 | 13.5 | 1819 | ≀ 5.3 | 79.2 | 15.6 |
| EN-control-CS-subjunctclause | Reference | 90 | 2.2 | 65.6 | 32.2 | 1130 | 4.0 | 70.7 | 25.3 |
| EN-control-CS-subjunctclause | UEDIN | 90 | 23.3 | ≀ 66.7 | 10.0 | 1130 | 22.4 | 60.7 | 16.9 |
| EN-control-CS-subjunctclause | CUNI Chimera | 90 | ≀ 21.1 | 58.9 | 20.0 | 1130 | 29.6 | 47.6 | 22.7 |
| EN-control-CS-subjunctclause | CUNI Chimera noDepFix | 90 | 21.1 | 58.9 | 20.0 | 1130 | 29.6 | 47.6 | 22.7 |
| EN-control-CS-subjunctclause | CUNI Transformer | 90 | ≀ 18.9 | ≀ 61.1 | 20.0 | 1130 | ≀ 21.7 | ≀ 57.5 | 20.8 |
| EN-control-CS-subjunctclause | online-B | 90 | 20.0 | 58.9 | 21.1 | 1130 | 25.3 | 52.7 | 21.9 |
| EN-control-CS-subjunctclause | online-A | 90 | 50.0 | 36.7 | 13.3 | 1130 | 52.8 | 30.4 | 16.7 |
| EN-control-CS-subjunctclause | online-G | 90 | 57.8 | 30.0 | 12.2 | 1130 | 64.6 | 20.4 | 15.0 |
| EN-control-CS-subjunctclause | CUNI Moses | 90 | 63.3 | 16.7 | 20.0 | 1130 | ≀ 61.6 | 17.1 | 21.3 |
| EN-gerund-CS-finclause | Reference | 75 | 25.3 | 56.0 | 18.7 | 165 | 21.2 | 59.4 | 19.4 |
| EN-gerund-CS-finclause | UEDIN | 75 | 26.7 | ≀ 58.7 | 14.7 | 165 | 28.5 | 57.0 | 14.5 |
| EN-gerund-CS-finclause | CUNI Chimera | 75 | ≀ 24.0 | ≀ 60.0 | 16.0 | 165 | ≀ 24.2 | 56.4 | 19.4 |
| EN-gerund-CS-finclause | CUNI Chimera noDepFix | 75 | 24.0 | 60.0 | 16.0 | 165 | 24.2 | 56.4 | 19.4 |
| EN-gerund-CS-finclause | CUNI Transformer | 75 | 28.0 | 50.7 | 21.3 | 165 | 26.7 | 51.5 | 21.8 |
| EN-gerund-CS-finclause | online-B | 75 | 34.7 | 48.0 | 17.3 | 165 | 32.7 | 47.3 | 20.0 |
| EN-gerund-CS-finclause | online-A | 75 | 60.0 | 26.7 | 13.3 | 165 | 57.6 | 32.7 | 9.7 |
| EN-gerund-CS-finclause | online-G | 75 | 61.3 | ≀ 28.0 | 10.7 | 165 | 63.6 | 27.9 | 8.5 |
| EN-gerund-CS-finclause | CUNI Moses | 75 | ≀ 38.7 | ≀ 45.3 | 16.0 | 165 | ≀ 36.4 | ≀ 48.5 | 15.2 |
| EN-gerund-CS-nofinclause | Reference | 218 | 0.5 | 72.9 | 26.6 | 368 | 2.2 | 70.1 | 27.7 |
| EN-gerund-CS-nofinclause | UEDIN | 218 | 16.1 | 67.4 | 16.5 | 368 | 19.8 | 64.9 | 15.2 |
| EN-gerund-CS-nofinclause | CUNI Chimera | 218 | 29.4 | 53.7 | 17.0 | 368 | 28.3 | 53.8 | 17.9 |
| EN-gerund-CS-nofinclause | CUNI Chimera noDepFix | 218 | 29.4 | 53.7 | 17.0 | 368 | 28.3 | 53.8 | 17.9 |
| EN-gerund-CS-nofinclause | CUNI Transformer | 218 | ≀ 17.0 | ≀ 62.8 | 20.2 | 368 | ≀ 16.6 | ≀ 63.0 | 20.4 |
| EN-gerund-CS-nofinclause | online-B | 218 | 19.3 | 59.6 | 21.1 | 368 | 20.7 | 58.2 | 21.2 |
| EN-gerund-CS-nofinclause | online-A | 218 | ≀ 12.8 | ≀ 75.7 | 11.5 | 368 | ≀ 14.7 | ≀ 72.8 | 12.5 |
| EN-gerund-CS-nofinclause | online-G | 218 | 17.4 | 71.6 | 11.0 | 368 | 16.3 | 71.5 | 12.2 |
| EN-gerund-CS-nofinclause | CUNI Moses | 218 | 33.5 | 50.5 | 16.1 | 368 | 32.1 | 51.4 | 16.6 |

Table 1: Detailed results of automatic tests. Left: Manually refined set, Right: larger, pre-selected set. Systems sorted by average performance in our testsuite. "≀" indicates lines ouf of sequence in the "Bad" or "Good" columns.

| System | BLEU | BLEU-cased | TER | BEER 2.0 | CharactTER |
|---|---|---|---|---|---|
| CUNI Transformer | 26.6 | 26.0 | 0.638 | 0.567 | 0.532 |
| UEDIN | 24.0 | 23.4 | 0.666 | 0.554 | 0.550 |
| CUNI Chimera noDepFix | 21.0 | 19.8 | 0.703 | 0.528 | 0.600 |
| CUNI Chimera | 20.8 | 19.2 | 0.704 | 0.522 | 0.605 |
| CUNI Moses | 17.5 | 16.4 | 0.739 | 0.509 | 0.632 |

Table 2: Automatic results of WMT18 English-Czech systems. From `matrix.statmt.org`.

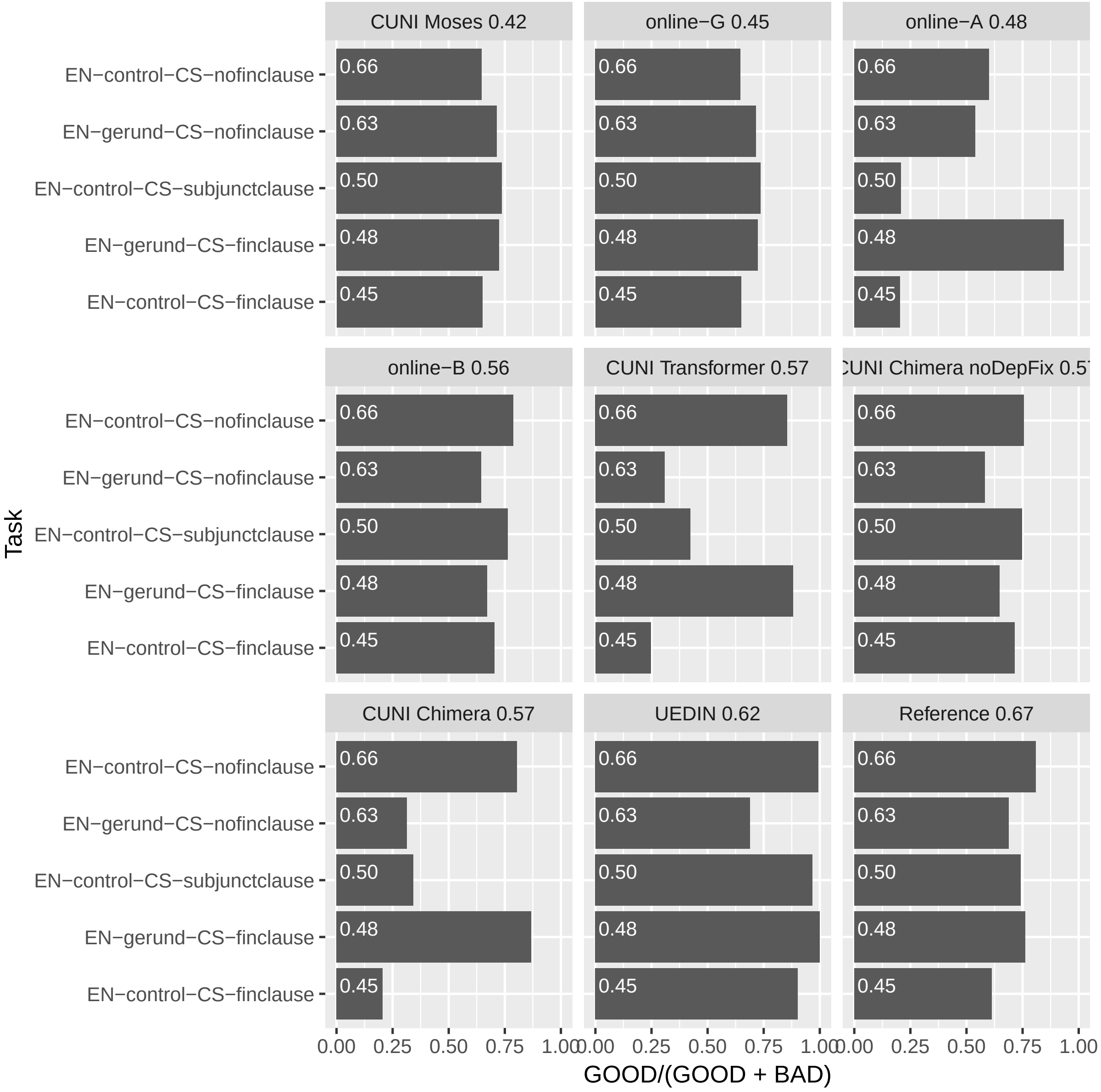


Figure 2: Performance by systems on the refined set. Each facet represents one MT system; each bar represents one pair of En phenomenon – Cs translation option. The result is computed as the proportion of agreements of the given MT system with the reference in the total number of cases (x-scale). In addition, we display the exact number inside each bar.

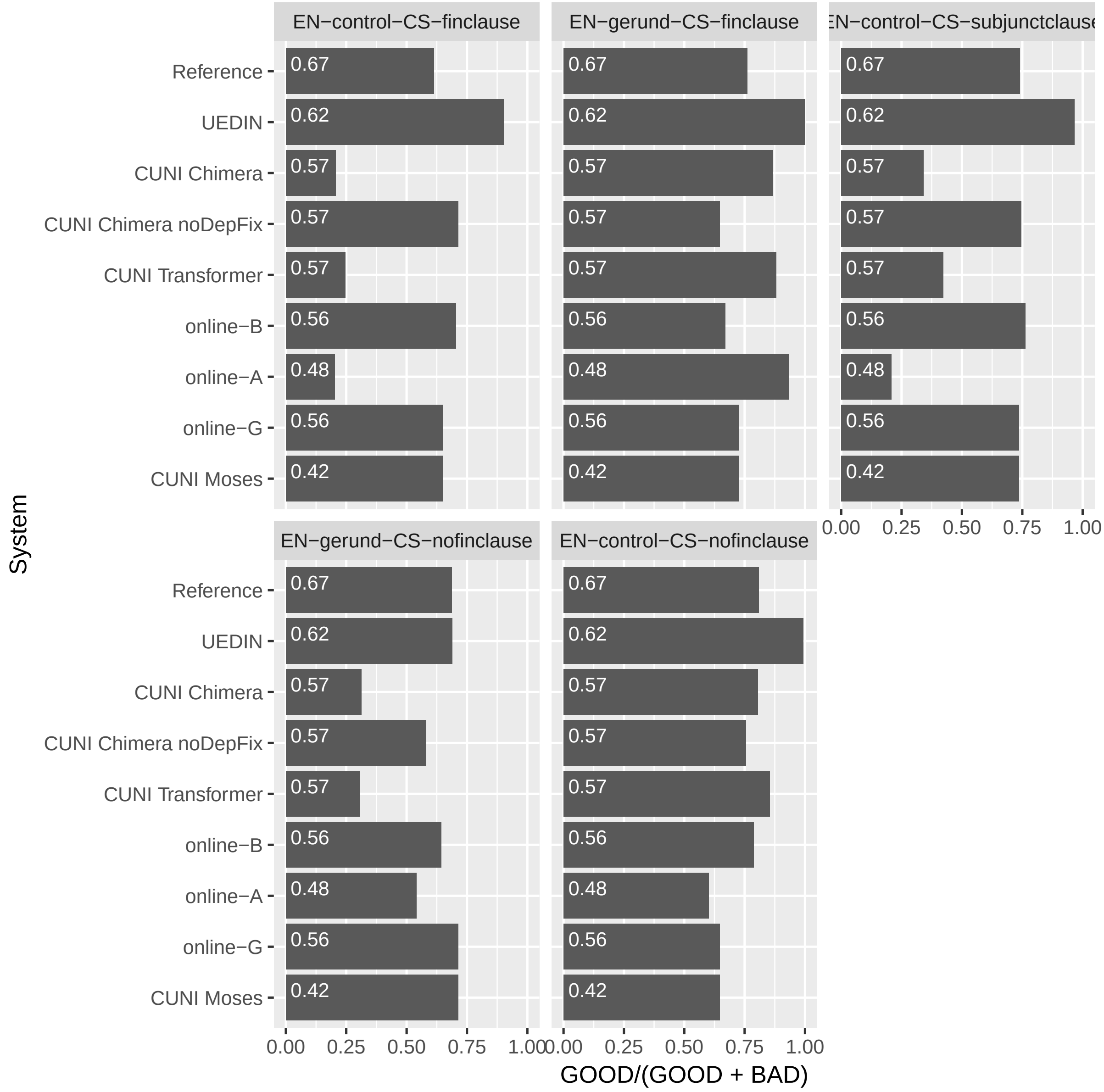


Figure 3: Performance by linguistic phenomena on the refined set. Each facet represents one pair of En phenomenon – Cs translation option. Each bar represents one MT system. The result is computed as the proportion of agreements of the given MT system with the reference in the total number of cases (x-scale). In addition, we display the exact number inside each bar.

# A Pronoun Test Suite Evaluation of the English–German MT Systems at WMT 2018

Liane Guillou[1*] Christian Hardmeier[2*]
Ekaterina Lapshinova-Koltunski[3*] Sharid Loáiciga[4*]
[1]School of Informatics, University of Edinburgh
[2]Department of Linguistics and Philology, Uppsala University
[3]Department of Language Science and Technology, Saarland University
[4]CLASP, University of Gothenburg
lguillou@inf.ed.ac.uk christian.hardmeier@lingfil.uu.se
e.lapshinova@mx.uni-saarland.de sharid.loaiciga@gu.se

## Abstract

We evaluate the output of 16 English-to-German MT systems with respect to the translation of pronouns in the context of the WMT 2018 competition. We work with a test suite specifically designed to assess system quality in various fine-grained categories known to be problematic. The main evaluation scores come from a semi-automatic process, combining automatic reference matching with extensive manual annotation of uncertain cases. We find that current NMT systems are good at translating pronouns with intra-sentential reference, but the inter-sentential cases remain difficult. NMT systems are also good at the translation of event pronouns, unlike systems from the phrase-based SMT paradigm. No single system performs best at translating all types of anaphoric pronouns, suggesting unexplained random effects influencing the translation of pronouns with NMT.

## 1 Introduction

Data-driven machine translation (MT) systems are very good at making translation choices based on the words in the immediate neighbourhood of the word currently being generated, but aspects of translation that require keeping track of long-distance dependencies continue to pose problems. Linguistically, long-distance dependencies often arise from discourse-level phenomena such as pronominal reference, lexical cohesion, text structure, etc. Initially largely ignored, such problems have attracted increasing attention in the statistical MT (SMT) community in recent years (Hardmeier, 2012; Sim Smith, 2017). One important problem that has proved to be surprisingly difficult despite extensive research is the translation of pronouns (Hardmeier et al., 2015; Guillou et al., 2016; Loáiciga et al., 2017).

Since the invention of the BLEU score (Papineni et al., 2002), the MT community has measured progress to a large extent with the help of summary scores that are easy to compute, but strongly affected by the corpus-level frequency of certain phenomena, and that tend to neglect specific linguistic relations and problems that occur infrequently. The advent of neural MT (NMT) with its improved capacity for modeling more complex relationships between linguistic elements has brought an increased interest in linguistic problems perceived as difficult, which are often not captured well by metrics like BLEU. It has been suggested that test suites composed of difficult cases could provide more relevant insights into the performance of MT systems than corpus-level summary scores (Hardmeier, 2015). In this paper, we present a semi-automatic evaluation of the systems participating in the English–German news translation track of the MT shared task at the WMT 2018 conference.

The analysis was carried out with the help of an English–German adaptation of the PROTEST test suite for pronoun translation (Guillou and Hardmeier, 2016). The test suite allows us to perform a fine-grained evaluation for different types of pronouns. Whilst the translation of event pronouns, which caused serious problems in earlier evaluations of SMT systems (Hardmeier et al., 2015; Hardmeier and Guillou, 2018), seems to be handled fairly well by modern NMT systems, we find that translating anaphoric pronouns is still difficult, especially (but not only) if the pronoun has an antecedent in a different sentence. Our results also confirm earlier findings that suggested the need for a careful evaluation that is sensitive to specific linguistic problems. Whilst BLEU scores as a measure of general translation quality are strongly correlated with pronoun correctness, there are significant outliers that would be missed by an evaluation focusing on BLEU only. Moreover, evaluating pro-

[*] All authors contributed equally.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 570–577
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64062

noun translations by comparison with a reference translation is not reliable for all types of pronouns (Guillou and Hardmeier, 2018). This fact limits the usefulness of automatic pronoun evaluation metrics such as APT (Miculicich Werlen and Popescu-Belis, 2017) and affects the semi-automatic evaluation of our test suite as well.

## 2 Related Work

Research on pronoun translation was boosted by three past shared tasks (Hardmeier et al., 2015; Guillou et al., 2016; Loáiciga et al., 2017). They focused on English, French, German and Spanish in different directions. To avoid the effort and cost of manual evaluation, the tasks were designed and evaluated as classification rather than MT tasks, except for the first year, which featured both MT and classification tasks. At the time of the first of these shared tasks, phrase-based SMT systems were still competitive and the winning system was a strong n-gram language model (not involving any translation) trained as a baseline. By the time of the last pronoun focused shared task, however, an NMT system with no explicit knowledge about pronouns ranked first (Jean et al., 2017).

Automatic metrics computed by matching the candidate and reference translations offer little explanation of the causes for error. Additionally, the neural architectures of current end-to-end systems make it difficult to find out where exactly a translation went wrong by inspection. Test suites ease the evaluation process in general, since they allow us to simultaneously measure quantitative performance and diagnose qualitative shortcomings with regard to the targeted set of problems.

Test suites assessing NMT have focused on contrastive pairs or sets of sentences automatically generated. These include Burlot and Yvon (2017), for the evaluation of morphology in the English-to-Latvian and to-Czech language pairs; Sennrich (2017), who evaluates noun phrase and subject-verb agreement, particle verbs, polarity, and transliteration; and Rios Gonzales et al. (2017) whose work concentrates on word sense disambiguation for the German-to-English and German-to-French pairs. The test suite used in our work is based on the PROTEST test suite, which was originally created for English–French by Guillou and Hardmeier (2016). Closest to our work is the test suite of English-to-French anaphoric pronouns and coherence and cohesion by Bawden et al. (2018). Their test suite includes 50 examples of contrastive pairs of sentences, which are manually created and targeted towards object pronouns.

## 3 Test Suite Construction

The data for our test suite was taken from the ParCorFull corpus (Lapshinova-Koltunski et al., 2018), a German-English parallel corpus manually annotated for co-reference. Although the corpus is designed for nominal co-reference, it includes annotations of two types of antecedents: entities and events. Entities can be either pronouns or noun phrases, whereas events can be verb phrases, clauses, or a set of clauses.

ParCorFull includes texts from TED talks transcripts and newswire data. Specifically, it includes the datasets used in the ParCor corpus (Guillou et al., 2014), the DiscoMT workshop (Hardmeier et al., 2016), and the test sets from the WMT 2017 shared task (Bojar et al., 2017).

We constructed a test suite of 200 pronoun translation examples for English–German with a focus on the ambiguous English pronouns *it* and *they* and the aim of providing a set of examples that represents the different problems machine translation researchers should consider. We extracted the examples from the TED talks section of ParCorFull.

The selection is based on a two-level hierarchy which considers pronoun *function* at the top level, followed by other pronoun attributes at the more granular lower level (for anaphoric pronouns only).

The English pronoun *they* functions as an anaphoric pronoun, whereas *it* can function as either an anaphoric (1), pleonastic (2), or event reference[1] pronoun (3), with each function requiring the use of different pronouns in German.

(1) a. The infectious disease that's killed more humans than any other is malaria. **It**'s carried in the bites of infected mosquitos.
b. Jene Krankheit, die mehr Leute als jede andere umgebracht hat, ist Malaria gewesen. **Sie** wird über die Stiche von infizierten Moskitos übertragen.

(2) a. And **it** seemed to me that there were three levels of acceptance that needed to take place.
b. Und **es** schien, dass es drei Stufen der Akzeptanz gibt, die alle zum Tragen kom-

[1] *Event* reference is more commonly known as abstract anaphora or discourse deixis.

men mussten.

(3) a. But I think if we lost everyone with Down syndrome, **it** would be a catastrophic loss.
b. Aber, wenn wir alle Menschen mit Down-Syndrom verlören, wäre **das** ein katastrophaler Verlust.

At the more granular lower level, anaphoric pronouns are subdivided according to the following attributes: whether the pronoun appears in the same sentence as its antecedent (intra-sentential) or a different sentence (inter-sentential), the antecedent is a group noun, the pronoun is in subject or non-subject position (*it* only), or an instance of *they* is used as a singular pronoun (for example, to refer to a person of unknown gender). An overview of the resulting categories is provided in Table 2.

The distribution of test suite examples over the pronoun categories in the hierarchy can be found in the first row of Table 3. The number of examples assigned to each category reflects a) the functional ambiguity of the pronoun *it*, b) the number of different translation options possible in German, and c) the number of pronouns in the corpus that belong to the category (for example, there are very few instances of *singular they* available). Within each category, we aim to create a balance in terms of the expected pronoun translation token. We achieve this by considering the translation of the set of possible candidates in the reference translation.

## 4 Evaluation Results

The evaluation included 10 systems submitted to the English–German sub-task of the WMT 2018 competition and 6 anonymized online translation systems. Among the WMT submissions, all of the systems are neural models, with the Transformer (Vaswani et al., 2017) being a popular architecture choice. Implementation details can be found in the system description papers published at WMT 2018.

### 4.1 Automatic Evaluation

We provide scores from two different automatic evaluation metrics for all systems in our dataset (see Table 1 and Figure 1). To give a general impression of the translation quality achieved by the various systems, we include the BLEU scores on the TED talks from which the test suite is derived. These scores differ from the BLEU scores of the official WMT evaluation because they are computed on a different test set, containing texts from a different domain. For a more pronoun-specific evaluation, we also compute APT scores (Miculicich Werlen and Popescu-Belis, 2017).[2] For better comparability, the set of pronouns evaluated by APT was restricted to the 200 items included in the test suite. Following the recommendations of Guillou and Hardmeier (2018), we did not define any "equivalent" pronouns in the APT metric, but counted exact matches only.

| System | BLEU | APT |
|---|---|---|
| Microsoft-Marian | 32.6 3 | 66.0 7 |
| NTT | 31.8 7 | **70.0** 1 |
| UCAM | 32.3 5 | 69.0 2 |
| uedin | 30.7 9 | 68.0 4 |
| MMT-prod | **33.2** 1 | 65.0 8 |
| KIT | 31.6 8 | 68.5 3 |
| online-Z | 32.5 4 | 66.5 6 |
| online-B | 32.7 2 | 62.5 10 |
| online-Y | 31.9 6 | 68.0 5 |
| JHU | 28.8 10 | 62.0 12 |
| online-F | 18.8 14 | 60.5 13 |
| LMU-nmt | 28.5 11 | 63.0 9 |
| online-A | 27.4 12 | 62.0 11 |
| online-G | 22.3 13 | 59.5 14 |
| RWTH-UNS | 13.7 15 | 54.5 15 |
| LMU-uns | 10.5 16 | – |

Table 1: Automatic evaluation results.

A regression fit between the BLEU scores obtained and the number of examples annotated as correct by each system indicates a strong correlation between the two (Figure 2; $r = 0.912$, $N = 16$, $p < 0.001$), as does a similar analysis for the APT score ($r = 0.887$, $N = 15$, $p < 0.001$). These results, however, should be taken with a grain of salt, as we argue further in Section 5.

### 4.2 Semi-automatic Evaluation

The semi-automatic evaluation method is a two-pass procedure. It is motivated by the observation that automatic reference-based methods can identify correct examples with relatively high precision, but low recall (Guillou and Hardmeier, 2018). The evaluation procedure relies on word alignments, which were generated automatically by running Giza++ (Och and Ney, 2003) in both directions with grow-diag-final symmetrization (Koehn et al., 2005). The word alignments for the examples in the reference translation were corrected manually.

In the first step, the candidate translations are matched against the reference translation to ap-

[2] The APT score could not be computed for the LMU-uns system because the scorer cannot handle completely untranslated sentences, which occur occasionally in the output of that system.

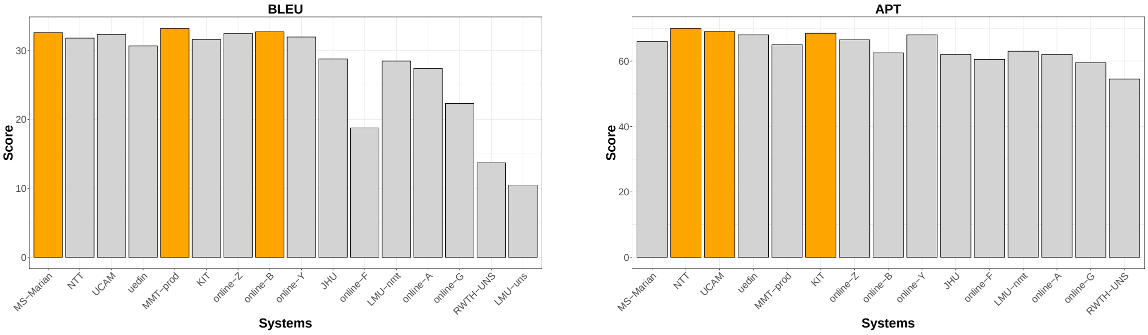


Figure 1: BLEU and APT scores. The three highest ranking systems are highlighted in orange.

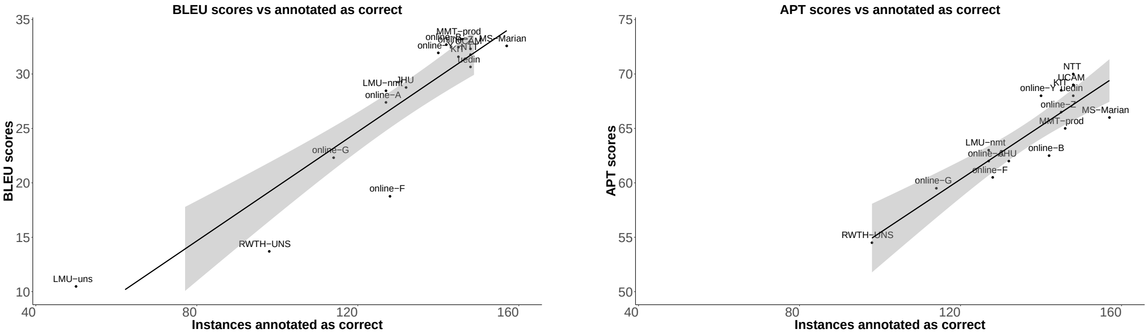


Figure 2: Correlation between the BLEU and APT scores and the number of instances annotated as correct. The gray zone indicates a 95% confidence interval.

prove examples that we can assume to be correct with reasonable confidence. Examples in the event and pleonastic categories can be approved based on a pronoun match alone; for the anaphoric categories, we also require matching antecedent translations. Two pronoun translations are considered to match if the sets of words aligned to the pronouns have at least one element in common after lowercasing. For antecedent translations, the word sequences aligned to the source antecedent must be completely equal for an automatic match. As a special exception, no automatic matches are generated for pronoun translations containing the word *sie* alone, so that the ambiguity between third-person plural *sie* and the pronoun of polite address *Sie* can be manually resolved.

In the second step, all examples not automatically approved are loaded into a graphical analysis tool specifically designed for the PROTEST test suite (Hardmeier and Guillou, 2016). The tool presents the annotator with the source pronoun, its translation by a given system, and the previous sentence for context. In the case of anaphoric pronouns, the context includes the sentence with the antecedent and one additional sentence. The examples were split randomly over four annotators. The annotators, who are translator trainees at Saarland University, are all native speakers of German with a good knowledge of English. To improve the quality of the annotations, the annotators had been trained beforehand on the output of a baseline NMT system.

| **Category** | **–** | **+** | **total** | **correct** |
|---|---|---|---|---|
| Anaphoric | | | | |
| intra-sent subj. *it* | 5 | 39 | 44 | 88.6% |
| intra-sent non-subj. *it* | 6 | 13 | 19 | 68.4% |
| inter-sent subj. *it* | 13 | 16 | 29 | 55.2% |
| inter-sent non-subj. *it* | 9 | 21 | 30 | 70.0% |
| intra-sent *they* | – | – | – | – |
| inter-sent *they* | – | – | – | – |
| singular *they* | – | – | – | – |
| group *it/they* | – | 9 | 9 | 100.0% |
| Event reference *it* | 14 | 68 | 82 | 82.9% |
| Pleonastic *it* | – | 137 | 137 | 100.0% |
| Total | 47 | 303 | 350 | 86.6% |

Table 2: Human evaluation of automatically approved examples

In total, 3,200 pronoun examples from 16 systems were evaluated. 1,150 examples were approved automatically and 2,050 examples were referred for manual annotation. To verify the validity of the semi-automatic method, we also solicited manual annotations for a random sample of 350 examples that had been approved automatically.

The first step of our two-step procedure can only approve examples, it never rejects them automatically. As a consequence, our semi-automatic evaluation is *biased towards correctness* with respect to a fully manual evaluation. The scores presented in Table 3 will therefore tend to overestimate the actual system performance.

The results of the human annotation of the random sample of 320 examples automatically matched as correct are presented in Table 2. Consistently with similar results for French (Hardmeier and Guillou, 2018), 86.6% of the automatically approved examples were accepted as correct by the evaluators. However, we must highlight that the accuracy of the automatic evaluation varies substantially across categories. Whilst pronouns known to be pleonastic can be checked automatically with very good confidence, the automatic evaluation of anaphoric pronouns is much more difficult, with an evaluation accuracy as low as 55.2% in the inter-sentential subject *it* case. This reflects the general difficulty of automatic pronoun evaluation (Guillou and Hardmeier, 2018) and reinforces the positive bias discussed in the previous paragraph for these categories in particular.

The results of the semi-automatic evaluation are displayed in Table 3. For the counts in this table, we used *manual* annotations wherever possible. Automatic annotations were used only for those examples that had not been annotated manually.

The best result was obtained by the Microsoft-Marian system, which translated 157 out of 200 pronouns correctly. It is followed by a group of 5 shared task submissions that achieved scores between 145 and 148. Three of the online systems also reached scores over 140. The remaining shared task submissions are JHU with a score of 132 and LMU-nmt with a score of 127. Unsurprisingly, the unsupervised submissions are ranked last.

## 5 Discussion

Generally speaking, a high BLEU score indicates good translation quality and vice versa. The APT score has been shown to capture good pronoun translations with reasonable precision, if unsatisfactory recall (Guillou and Hardmeier, 2018), but it is also trivially correlated with our test suite score to some extent because the automatic part of our semi-automatic evaluation identifies good translation with a mechanism that is very similar to that of APT. In the right half of Figure 2, we observe that the APT score introduces spurious differences between systems reaching exactly the same number of correctly translated items (NTT, UCAM, uedin) and fails to reward correct pronoun translations in some of the systems (Microsoft-Marian, online-B). As a result, the score can serve as an indicator, but not as a reliable replacement of a manual or semi-automatic evaluation.

Moreover, the small size of the test suite and the differences between the system architectures must be kept in mind. Considering these two factors, a larger threshold in any of the two scores is needed to claim that one system is actually better than another (Berg-Kirkpatrick et al., 2012). This caveat appears to be confirmed by the two outliers seen in the left part of Figure 2. Interestingly, the online-F system achieves many good pronoun translations despite a low BLEU score. The RWTH-uns system is also much better on correct pronouns than LMU-uns (the other unsupervised system) than the difference in BLEU scores would suggest.

The results of manual evaluation vary significantly by category. In the anaphoric *it* categories, it is evident that intra-sentential anaphora is easier to handle than inter-sentential anaphora. In the intra-sentential case, the best systems produce correct translations for 70–80% of the examples, which is a fair result, but indicates that the problem is not completely solved yet. In the inter-sentential *it* categories, the average performance is below 50% despite the positive bias of our evaluation method, and even the best-performing systems are not much better. It is worth noting that no single system performs best over all anaphoric categories, which suggests that the top scores achieved for this part of the test suite could be random strokes of luck. The results for pronouns in subject and non-subject positions are not very different. This contrasts with the results of Hardmeier and Guillou (2018) for English–French, where non-subject pronouns were found to be substantially harder to translate. It might be due to the fact that the direct object forms of French personal pronouns coincide with those of the definite article, a problem that does not apply to German.

The plural cases of *they* do not cause any serious problems, at least for the stronger systems, since *they* can usually be translated straightforwardly using the German pronoun *sie*. The errors occurring in these categories are often due to confusion with the pronoun of polite address *Sie* (“you”). When

| | Pronouns | | | | | | | | | | | Antecedents |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | anaphoric | | | | | | | | event | pleonastic | | |
| | it | | | | they | | | it/they | it | it | | |
| | intra | | inter | | intra | inter | sing. | group | | | | |
| | subj. | non-subj. | subj. | non-subj. | | | | | | | Total | |
| *Examples* | *25* | *25* | *25* | *25* | *10* | *10* | *5* | *15* | *30* | *30* | ***200*** | *140* |
| Microsoft-Marian | 18 | 20 | 12 | 15 | 9 | 10 | 2 | 13 | 29 | 29 | **157** | 132 |
| NTT | 16 | 18 | 14 | 16 | 10 | 10 | 1 | 8 | 26 | 29 | **148** | 135 |
| UCAM | 19 | 20 | 13 | 11 | 10 | 10 | 2 | 11 | 22 | 30 | **148** | 134 |
| uedin | 19 | 19 | 10 | 11 | 10 | 10 | – | 11 | 29 | 29 | **148** | 132 |
| MMT-prod | 20 | 19 | 11 | 15 | 10 | 8 | – | 9 | 25 | 29 | **146** | 137 |
| KIT | 19 | 18 | 15 | 11 | 9 | 9 | 1 | 6 | 27 | 30 | **145** | 126 |
| online-Z | 21 | 18 | 10 | 10 | 10 | 10 | 2 | 11 | 24 | 29 | **145** | 132 |
| online-B | 20 | 15 | 12 | 12 | 8 | 10 | – | 8 | 27 | 30 | **142** | 128 |
| online-Y | 18 | 17 | 11 | 12 | 10 | 9 | 1 | 8 | 24 | 30 | **140** | 136 |
| JHU | 12 | 17 | 8 | 11 | 8 | 10 | 3 | 10 | 24 | 29 | **132** | 119 |
| online-F | 13 | 16 | 10 | 11 | 10 | 10 | 2 | 7 | 21 | 28 | **128** | 115 |
| LMU-nmt | 10 | 9 | 10 | 13 | 7 | 10 | 1 | 9 | 28 | 30 | **127** | 125 |
| online-A | 11 | 9 | 12 | 16 | 5 | 10 | 2 | 5 | 27 | 30 | **127** | 130 |
| online-G | 10 | 6 | 15 | 11 | 2 | 8 | 2 | 7 | 23 | 30 | **114** | 119 |
| RWTH-uns | 9 | 5 | 9 | 8 | 3 | 8 | 1 | 7 | 19 | 29 | **98** | 99 |
| LMU-uns | 4 | 2 | 2 | 2 | 4 | 8 | – | 5 | 15 | 8 | **50** | 87 |
| *Average* | | | | | | | | | | | | |
| count | 14.9 | 14.3 | 10.9 | 11.6 | 7.8 | 9.4 | 1.3 | 8.4 | 24.4 | 28.0 | **130.9** | 124.1 |
| percentage | 59.8 | 57.0 | 43.5 | 46.3 | 78.1 | 93.8 | 25.0 | 56.3 | 81.3 | 93.5 | **65.4** | 88.6 |

Table 3: Pronoun and antecedent translations marked as correct, per system

*they* has a singular antecedent or refers to a group, however, it is mistranslated much more frequently.

The only system that has noticeable problems with pleonastic *it* is the unsupervised LMU-uns submission. Translating event *it* seems to be more difficult, but many systems still achieve close to perfect results in this category. Similarly to the results of Hardmeier and Guillou (2018) for English–French, this suggests that NMT systems are quite good at identifying pronouns with event reference and producing appropriate translations for them.

## 6 Conclusions

We have presented a detailed analysis of 16 NMT systems, assessing their performance in the translation of pronouns using a semi-automatic evaluation based on a balanced test suite. The results reinforce the idea that automatic evaluation scores are correlated with manual evaluation results, but they also confirm that automatic evaluation can provide a misleading picture of the behavior of some systems. The evaluation has also reinforced that special attention should be paid to the problematic cases that are only identifiable through the careful balance of categories achieved in the test suite design. This balanced design has also made us aware of the progress made by NMT in the modeling of context for the translation of pleonastic, event and intra-sentential anaphoric pronouns. Pleonastic pronouns are handled almost perfectly by most systems, so we suggest that future evaluations emphasize the more challenging cases. Anaphoric pronouns depending on the inter-sentential context remain a significant challenge. They present an ideal test case for the development of context-aware NMT systems. Research in that direction has recently gained some traction (Tiedemann and Scherrer, 2017; Wang et al., 2017; Tu et al., 2018) and has claimed promising results specifically for pronoun translation (Voita et al., 2018). It remains to be seen whether the development of such methods will lead to a breakthrough in the translation of inter-sentential anaphoric pronouns in the near future.

## Acknowledgements

The work carried out at Uppsala University was supported by the Swedish Research Council under grants 2012-916 (to Jörg Tiedemann) and 2017-930 (to Christian Hardmeier). We thank Jörg Tiedemann for helping us fund this effort. The work carried out at The University of Edinburgh was funded by the ERC H2020 Advanced Fellowship GA 742137 SEMANTAX and a grant from The

University of Edinburgh and Huawei Technologies. The manual test suite evaluation was funded by the European Association for Machine Translation. We thank our annotators Daria Hert, Georg Seiler, Alexander Schütz and Peter Schneider for their valuable work.

## References

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Franck Burlot and François Yvon. 2017. Evaluating the morphological competence of machine translation systems. In *Proceedings of the Second Conference on Machine Translation*, pages 43–55, Copenhagen, Denmark. Association for Computational Linguistics.

Liane Guillou and Christian Hardmeier. 2016. PROTEST: A test suite for evaluating pronouns in machine translation. In *Proceedings of the Eleventh Language Resources and Evaluation Conference*, LREC 2016, pages 636–643, Portorož, Slovenia.

Liane Guillou and Christian Hardmeier. 2018. Automatic reference-based evaluation of pronoun translation misses the point. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP, Brussels, Belgium. Association for Computational Linguistics.

Liane Guillou, Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, Mauro Cettolo, Bonnie Webber, and Andrei Popescu-Belis. 2016. Findings of the 2016 WMT shared task on cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation*, pages 525–542, Berlin, Germany. Association for Computational Linguistics.

Liane Guillou, Christian Hardmeier, Aaron Smith, Jörg Tiedemann, and Bonnie Webber. 2014. ParCor 1.0: A parallel pronoun-coreference corpus to support statistical MT. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, LREC 2014, pages 3191–3198, Reykjavik, Iceland. European Language Resources Association (ELRA).

Christian Hardmeier. 2012. Discourse in statistical machine translation: A survey and a case study. *Discours*, 11.

Christian Hardmeier. 2015. On statistical machine translation and translation theory. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 168–172, Lisbon (Portugal). Association for Computational Linguistics.

Christian Hardmeier and Liane Guillou. 2016. A graphical pronoun analysis tool for the protest pronoun evaluation test suite. *Baltic Journal of Modern Computing*, 4(2):318–330.

Christian Hardmeier and Liane Guillou. 2018. Pronoun translation in English–French machine translation: An analysis of error types. *ArXiv e-prints*, 1808.10196.

Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, and Mauro Cettolo. 2015. Pronoun-focused MT and cross-lingual pronoun prediction: Findings of the 2015 DiscoMT shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16, Lisbon, Portugal. Association for Computational Linguistics.

Christian Hardmeier, Jörg Tiedemann, Preslav Nakov, Sara Stymne, and Yannick Versely. 2016. DiscoMT 2015 shared task on pronoun translation. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague. `http://hdl.handle.net/11372/LRT-1611`.

Sébastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Neural machine translation for cross-lingual pronoun prediction. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 54–57, Copenhagen, Denmark. Association for Computational Linguistics.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania.

Ekaterina Lapshinova-Koltunski, Christian Hardmeier, and Pauline Krielke. 2018. ParCorFull: a parallel corpus annotated with full coreference. In *Proceedings of 11th Language Resources and Evaluation Conference*, pages 423–428, Miyazaki, Japan. European Language Resources Association (ELRA).

Sharid Loáiciga, Sara Stymne, Preslav Nakov, Christian Hardmeier, Jörg Tiedemann, Mauro Cettolo, and Yannick Versley. 2017. Findings of the 2017 DiscoMT shared task on cross-lingual pronoun prediction. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 1–16, Copenhagen, Denmark. Association for Computational Linguistics.

Lesly Miculicich Werlen and Andrei Popescu-Belis. 2017. Validation of an automatic metric for the accuracy of pronoun translation (APT). In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 17–25, Copenhagen, Denmark. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ACL 2002, pages 311–318, Philadelphia. Association for Computational Linguistics.

Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? Assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain. Association for Computational Linguistics.

Karin Sim Smith. 2017. On integrating discourse in machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 110–121, Copenhagen, Denmark. Association for Computational Linguistics.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics*, 6:407–420.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark. Association for Computational Linguistics.

# Fine-grained evaluation of German-English Machine Translation based on a Test Suite

**Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, Hans Uszkoreit**
German Research Center for Artificial Intelligence (DFKI), Berlin, Germany
firstname.lastname@dfki.de

## Abstract

We present an analysis of 16 state-of-the-art MT systems on German-English based on a linguistically-motivated test suite. The test suite has been devised manually by a team of language professionals in order to cover a broad variety of linguistic phenomena that MT often fails to translate properly. It contains 5,000 test sentences covering 106 linguistic phenomena in 14 categories, with an increased focus on verb tenses, aspects and moods. The MT outputs are evaluated in a semi-automatic way through regular expressions that focus only on the part of the sentence that is relevant to each phenomenon. Through our analysis, we are able to compare systems based on their performance on these categories. Additionally, we reveal strengths and weaknesses of particular systems and we identify grammatical phenomena where the overall performance of MT is relatively low.

## 1 Introduction

The evaluation of Machine Translation (MT) has mostly relied on methods that produce a numerical judgment on the correctness of a test set. These methods are either based on the human perception of the correctness of the MT output (Callison-Burch et al., 2007), or on automatic metrics that compare the MT output with the reference translation (Papineni et al., 2002; Snover et al., 2006). In both cases, the evaluation is performed on a test-set containing articles or small documents that are assumed to be a random representative sample of texts in this domain. Moreover, this kind of evaluation aims at producing average scores that express a generic sense of correctness for the entire test set and compare the performance of several MT systems.

Although this approach has been proven valuable for the MT development and the assessment of new methods and configurations, it has been suggested that a more fine-grained evaluation, associated with linguistic phenomena, may lead in a better understanding of the errors, but also of the efforts required to improve the systems (Burchardt et al., 2016). This is done through the use of test suites, which are carefully devised corpora, whose test sentences include the phenomena that need to be tested. In this paper we present the fine-grained evaluation results of 16 state-of-the-art MT systems on German-English, based on a test suite focusing on 106 German grammatical phenomena with a focus on verb-related phenomena.

## 2 Related Work

The use of test suites in the evaluation of NLP applications (Balkan et al., 1995) and MT systems in particular (King and Falkedal, 1990; Way, 1991) has been proposed already in the 1990's. For instance, test suites were employed to evaluate state-of-the-art rule-based systems (Heid and Hildenbrand, 1991). The idea of using test suites for MT evaluation was revived recently with the emergence of Neural MT (NMT) as the produced translations reached significantly better levels of quality, leading to a need for more fine-grained qualitative observations. Recent works include test suites that focus on the evaluation of particular linguistic phenomena (e.g. pronoun translation; Guillou and Hardmeier, 2016) or more generic test suites that aim at comparing different MT technologies (Isabelle et al., 2017; Burchardt et al., 2017) and Quality Estimation methods (Avramidis et al., 2018). The previously presented papers differ in the amount of phenomena and the language pairs they cover.

This paper extends the work presented in Burchardt et al. (2017) by including more test sentences and better coverage of phenomena. In con-

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 578–587
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64063

trast to that work, which applied the test suite in order to compare 3 different types of MT systems (rule-based, phrase-based and NMT), the evaluation in the publication at hand has been applied on 16 state-of-the-art systems whose majority follows the NMT methods.

## 3 Method

This test suite is a manually devised test set, aiming to investigate the MT performance against a wide range of linguistic phenomena or other qualitative requirements (e.g. punctuation).

It contains a set of sentences in the source language, written or chosen by a team of linguists and professional translators with the aim to cover as many linguistic phenomena as possible, and particularly the ones that MT often fails to translate properly. Each sentence of the test suite serves as a paradigm for investigating only one particular phenomenon. Given the test sentences, the evaluation tests the ability of the MT systems to properly translate the associated phenomena. The phenomena are organized in categories (e.g. although each verb tense is tested separately with the respective test sentences, the results for all tenses are aggregated in the broader category of verb tense/aspect/mood).

Our test suite contains about 5,000 test sentences, covering 106 phenomena organized in 14 categories. For each phenomenon at least 20 test sentences were devised to allow better generalizations about the capabilities of the MT systems. With 88%, the majority of the test suite covers verb phenomena, but other categories, such as negation, long distance dependencies, valency or multi-word expressions are included as well. A full list of the phenomena and their categories can be seen in Table 1. An example list of test sentences with correct and incorrect translations is available on GitHub[1].

### 3.1 Construction of the Test Suite

The test suite was constructed in a way that allows a semi-automatic evaluation method, in order to assist the efficient evaluation of many translation systems. A simplified sketch of the test suite construction is shown in Figure 1. First (Figure 1, stage a), the linguist choses or writes the test sentences in the source language with the help of translators. The test sentences are manually written or chosen, based on whether their translation has demonstrated or is suspected to demonstrate MT errors of the respective error categories. Test sentences are selected from various parallel corpora or drawn from existing resources, such as the TSNLP Grammar Test Suite (Lehmann et al., 1996) and online lists of typical translation errors. Then (stage b) the test sentences are passed as an input to the some sample MT systems and their translations are fetched.

Based on the output of the sample MT systems and the types of the errors, the linguist devises a set of hand-crafted regular expressions (stage c) while the translator ensures the correctness of the expressions. The regular expressions are used to automatically check if the output correctly translates the part of the sentence that is related to the phenomenon under inspection. There are regular expressions that match correct translations (positive) as well as regular expressions that match incorrect translations (negative).

### 3.2 Application of the Test Suite

During the evaluation phase, the test sentences are given to several translation systems and their outputs are acquired (stage d). The regular expressions are applied to the MT outputs (stage e) to automatically check whether the MT outputs translate the particular phenomenon properly. An MT output is marked as correct (*pass*), if it matches a positive regular expression. Similarly, it is marked as incorrect (*fail*), if it matches a negative regular expression. In cases where the MT output does not match either a positive or a negative regular expression, the automatic evaluation flags an uncertain decision (*warning*). Then, the results of the automatic annotation are given to a linguist or a translator who manually checks the warnings (stage f) and optionally refines the regular expressions in order to cover similar future cases. It is also possible to add full sentences as valid translations, instead of regular expressions. In this way, the test suite grows constantly, whereas the required manual effort is reduced over time.

Finally, for every system we calculate the phenomenon-specific translation accuracy:

$$\text{accuracy} = \frac{\text{correct translations}}{\text{sum of test sentences}}$$

The translation accuracy per phenomenon is given by the number of the test sentences for the pheno-

[1] https://github.com/DFKI-NLP/TQ_AutoTest

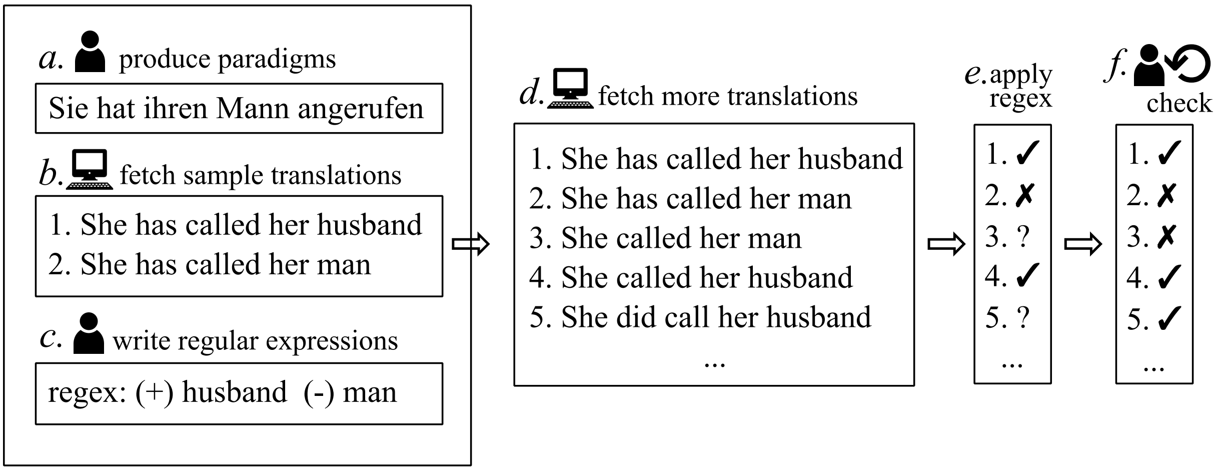

a. produce paradigms
Sie hat ihren Mann angerufen
b. fetch sample translations
1. She has called her husband
2. She has called her man
c. write regular expressions
regex: (+) husband (-) man
d. fetch more translations
1. She has called her husband
2. She has called her man
3. She called her man
4. She called her husband
5. She did call her husband
...
e. apply regex
1. ✓
2. ✗
3. ?
4. ✓
5. ?
...
f. check
1. ✓
2. ✗
3. ✗
4. ✓
5. ✓
...

Figure 1: Example of the preparation and application of the test suite for one test sentence

menon which were translated properly, divided by the number of all test sentences for this phenomenon.

This allows us also to perform comparisons among the systems, focusing on particular phenomena. The significance of every comparison between two systems is confirmed with a two-tailed Z-test with $\alpha = 0.95$, testing the null hypothesis that the difference between the two respective percentages is zero.

### 3.3 Experiment setup

The evaluation of the MT outputs was performed with TQ-AutoTest (Macketanz et al., 2018), a tool that organizes the test items in a database, allowing the application of the regular expressions on new MT outputs. For the purpose of this study, we have compared the 16 systems submitted to the test suite task of the EMNLP2018 Conference of Machine Translation (WMT18) for German→English. At the time that this paper is written, the creators of 11 of these systems have made their development characteristics available, 10 of them stating that they follow a NMT method and one of them a method combining phrase-based SMT and NMT.

After the application of the existing regular expressions to the outputs of these 16 systems, there was a considerable amount of warnings (i.e. uncertain judgments) that varied between 10% and 45% per system. A manual inspection of the outputs was consequently performed (Figure 1, stage f) by a linguist, who invested approximately 80 hours of manual annotation. A small-scale manual inspection of the automatically assigned *pass* and *fail* labels indicated that the percentage of the erroneously assigned labels is negligible. The manual inspection therefore focused on warnings and reduced their amount to less than 10% warnings per system[2]. In particular, 32.1% of the original system outputs ended in warnings, after the application of the regular expressions, whereas the manual inspection and the refining of the regular expressions additionally validated 14,000 of these system outputs, i.e. 15.7% of the original test suite.

In order to analyze the results with respect to the existence of warnings, we performed two different types of analysis:

1. Remove all sentences from the *overall comparison* that have even one warning for one system and the translation accuracy on the remaining segments. The unsupervised systems are completely excluded from this analysis in order to keep the sample big enough. This way, all systems are compared on the same set of segments.

2. Remove the sentences with warnings *per system* and calculate the translation accuracy on the remaining segments. The unsupervised systems can be included in this analysis. In this way, the systems are *not* compared on the same set of segments, but more segments can be included altogether.

## 4 Results

The final results of the evaluation can be seen in Table 2, based on Analysis 1 and Table 3, based

[2] Here, we do not take into account the two unsupervised systems for the reasons explained in Section 4.1.

on Analysis 2. Results for verb-related phenomena based on Analysis 1 are detailed in Tables 4 and 5 and other indicative phenomena in Table 6. The filtering prior to Analysis 1 left a small number of test sentences per category, which limits the possibility to identify significant differences between the systems. Analysis 2 allows better testing of each system's performance, but observations need to be treated with caution, since the systems are tested against different test sentences and therefore the comparisons between them are not as expressive as in Analysis 1. Moreover, the interpretability of the overall averages of these tables is limited, as the distribution of the test sentences and the linguistic phenomena does not represent an objective notion of quality.

We have calculated the mean values per system as non-weighted average and as weighted average. The non-weighted average was calculated by dividing the sum off all correct translations by the sum of all test sentences. The weighted average for a system was computed by taking a mean of the averages per category. We have not calculated statistical significances for the weighted averages as these are less meaningful due to the dominance of the verb tense/aspect/mood category.

### 4.1 Comparison between systems

The following results are based on Analysis 1. The system that achieves the highest accuracy in most linguistic phenomena, as compared to the rest of the systems, is UCAM, which is in the first significance cluster for 11 out of the 12 decisive error categories in Analysis 1 and achieves a 86.0% non-weighted average accuracy over all test sentences. UCAM obtains a significantly better performance than all other systems concerning verb tense/aspect/mood, reaching a 86.9% accuracy, 1.5% better than MLLP and NTT which are following in this category. The different performance may be explained by the fact that UCAM differs from others, since it combines several difference neural models together with a phrase-based SMT system in an syntactic MBR-based scheme (Stahlberg et al., 2016). Despite its good performance in grammatical phenomena, UCAM has a very low accuracy regarding punctuation (52.9%).

The system with the highest weighted average score is RWTH. Even though it reaches higher accuracies for some categories than UCAM, the differences are not statistically significant.

Another system that achieves the best accuracies at the 11 out of the 12 categories is Online-A. This system performs close to the average of all systems concerning verb tense/aspect/mood, but it shows a significantly better performance on the category of punctuation (96.1%). Then, 6 systems (JHU, NTT, Online-B, Online-Y, RWTH, Ubiqus) have the best performance at the same amount of categories (10 out of 12), having lost the first position in punctuation and verb tense/aspect/mood.

Two systems that have the lowest accuracies in several categories are Online-F and Online-G. Online-F has severe problems with the punctuation (3.9%) since it failed producing proper quotation marks in the output and mistranslated other phenomena, such as commas and the punctuation in direct speech (see Table 6). Online-G has the worst performance concerning verb tense/aspect/mood (45.8%). Additionally, these two systems together demonstrate the worst performance on coordination/ellipsis and negation.

The **unsupervised systems** form a special category of systems trained only on monolingual corpora. Their outputs suffer from adequacy problems, often being very "creative" or very far from a correct translation. Thus, the automatic evaluation failed to check a vast amount of test sentences on these systems. Therefore, we conducted Analysis 2. As seen in Table 3, unsupervised systems suffer mostly on MWE (11.1% - 17.4% accuracy), function words (15.7% - 21.7%), ambiguity (26.9% - 29.1%) and non-verbal agreement (38.3% - 39.6%).

### 4.2 Linguistic categories

Despite the significant progress in the MT quality, we managed to devise test sentences that indicate that the submitted systems have a mediocre performance for several linguistic categories. On average, all current state-of-the-art systems suffer mostly on punctuation (and particularly quotation marks), MWE, ambiguity and false friends with an average accuracy of less than 64% (based on Analysis 1). Verb tense/aspect/mood, nonverbal agreement, function words and coordination/ellipsis are also far from good, with average accuracies around 75%.

The two categories verb valency and named entities/terminology cannot lead to comparisons on the performance of individual systems, since all systems achieve equal or insignificantly different performance on them. The former has an average accuracy of 81.4%, while the latter has an average accuracy of 83.4%.

We would like to present a few examples in order to provide a better understanding of the linguistic categories and the evaluation. Example (1) is taken from the category of **punctuation**. Among others, we test the punctuation in the context of direct speech. While in German it is introduced by a colon, in English it is introduced by a comma. In this example, the NTT system produces a correct output (therefore highlighted in boldface), whereas the other two systems depict incorrect translations with a colon.

(1) Punctuation

| | |
|---|---|
| *source:* | *Er rief: „Ich gewinne!“* |
| **NTT:** | He shouted, “I win!” |
| Online-F: | He called: “I win!” |
| Ubiqus: | He cried: “I win!” |

We may assume that these errors are attributed to the fact that punctuation is often manipulated by hand-written pre- and post-processing tools, whereas the ability of the neural architecture to properly convey the punctuation sequence has attracted little attention and is rarely evaluated properly.

**Negation** is one of the most important categories for meaning preservation. Two commercial systems (Online-F and Online-G) show the lowest accuracy for this category and it is disappointing that they miss 4 out of 10 negations. In Example (2), the German negation particle “nie” should be translated as “never”, but Online-G omits the whole negation. In other cases it negates the wrong element in the sentence.

(2) Negation

| | |
|---|---|
| *source:* | *Tim wäscht seine Kleidung nie selber.* |
| **Online-B:** | Tim never washes his clothes himself. |
| Online-G: | Tim is washing his clothes myself. |

**MWE**, such as idioms or collocations, are prone to errors in MT as they cannot be translated in their separate elements. Instead, the meaning of the expression has to be translated as a whole. Example (3) focuses on the German idiom “auf dem Holzweg sein” which can be translated as “being on the wrong track”. However, a literal translation of “Holzweg” would be “wood(en) way”, “wood(en) track” or “wood(en) patch”. As can be seen in the example, MLLP and UCAM provide a literal translation of the separate segments of the MWE rather than translating the whole meaning of it, resulting in a translation error.

(3) MWE

| | |
|---|---|
| *source:* | *Du bist auf dem Holzweg.* |
| MLLP: | You’re on the wood track. |
| **RWTH:** | You’re on the wrong track. |
| UCAM: | You’re on the wooden path. |

### 4.3 Linguistic phenomena

As mentioned above, a large part of the test suite is made up of verb-related phenomena. Therefore, we have conducted a more fine-grained analysis of the category “Verb tense/aspect/mood”. In Table 4 we have grouped the phenomena by verb tenses. Table 5 shows the results for the verb-related phenomena grouped by verb type. Regarding the verb tenses, future II and future II subjunctive show the lowest accuracy with a maximum accuracy of about 30%. The highest accuracy value on average (weighted and non-weighted) is achieved by UCAM with 63.5%, respectively 61.5%. UCAM is the only system that is one of the best-performing systems for all the verb tenses as well as for all the verb types. The second-best system on average for verb tenses and verb types is NTT.

While the accuracy scores among the phenomena range between 33.4% and 63.5% for the verb tenses, the scores for the verb types are higher with 45.7% - 86.9%.

Table 6 shows interesting individual phenomena with at least 15 valid test sentences. The accuracy for compounds and location is generally quite high. There are other phenomena that exhibit a larger range of accuracy scores, as for example quotation marks, with an accuracy ranging from 0% to 94.7% among the systems. The system Online-F fails on all test sentences with quotation marks. The failure results from the system generating the quotation marks analogously to the German punctuation, e.g., introducing direct speech with a colon, as seen in Example (1). Online-F furthermore also fails on all test sentences with question tags, as does NJUNMT. For the phenomenon location, on the other hand, none of the systems is significantly better than any other system. They all perform similarly good, with an accuracy ranging from 86.7% to 100%. RWTH is the only system that reaches an accuracy of 100% twice in these selected phenomena.

## 5 Conclusion and Further Work

We used a test suite in order to perform fine-grained evaluation in the output of the state-of-the-art systems, submitted at the shared task of WMT18. One system (UCAM), that uses a syntactic MBR combination of several NMT and phrase-based SMT components, stands out regarding to verb-related phenomena. Additionally, two systems fail to translate 4 out of 10 negations. Generally, submitted systems suffer on punctuation (and particularly quotation marks, with the exception of Online-A), MWE, ambiguity and false friends, and also on translating the German future tense II. 6 systems have approximately the same performance in a big number of linguistic categories.

Fine-grained evaluation would ideally provide the potential to identify particular flaws at the development of the translation systems and suggest specific modifications. Unfortunately, at the time that this paper was written, few details about the development characteristics of the respective systems were available, so we could provide only few assumptions based on our findings. The differences observed may be attributed to the design of the models, to pre- and post-processing tools, to the amount, the type and the filtering of the corpora and other development decisions. We believe that the findings are still useful for the original developers of the systems, since they are aware of all their technical decisions and they have the technical possibility to better inspect the causes of specific errors.

## Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF) through the projects DEEPLEE (01lW17001) and UPLINX (01IS17066A).

Special thanks to Arle Lommel and Kim Harris who helped with their input in earlier stages of the experiment, to Renlong Ai and He Wang who developed and maintained the technical infrastructure and to Aylin Cornelius who helped with the evaluation.

## References

Eleftherios Avramidis, Vivien Macketanz, Arle Lommel, and Hans Uszkoreit. 2018. Fine-grained evaluation of Quality Estimation for Machine translation based on a linguistically-motivated Test Suite. In *Proceedings of the First Workshop on Translation Quality Estimation and Automatic Post-Editing*, pages 243–248, Boston, MA, USA.

Lorna Balkan, Doug Arnold, and Siety Meijer. 1995. Test suites for natural language processing. In *Aslib proceedings*, volume 47, pages 95–98. MCB UP Ltd.

Aljoscha Burchardt, Kim Harris, Georg Rehm, and Hans Uszkoreit. 2016. Towards a Systematic and Human-Informed Paradigm for High-Quality Machine Translation. In *Language Resources and Evaluation (LREC)*, Portoroz, Slovenia. European Language Resources Association.

Aljoscha Burchardt, Vivien Macketanz, Jon Dehdari, Georg Heigold, Jan-Thorsten Peter, and Philip Williams. 2017. A Linguistic Evaluation of Rule-Based, Phrase-Based, and Neural MT Engines. *The Prague Bulletin of Mathematical Linguistics*, 108:159–170.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (Meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic. Association for Computational Linguistics.

Liane Guillou and Christian Hardmeier. 2016. PROTEST: A Test Suite for Evaluating Pronouns in Machine Translation. *Tenth International Conference on Lan- guage Resources and Evaluation (LREC 2016)*.

Ulrich Heid and Elke Hildenbrand. 1991. Some practical experience with the use of test suites for the evaluation of SYSTRAN. In *the Proceedings of the Evaluators' Forum, Les Rasses*. Citeseer.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A Challenge Set Approach to Evaluating Machine Translation. In *EMNLP 2017: Conference on Empirical Methods in Natural Language Processing*.

Margaret King and Kirsten Falkedal. 1990. Using test suites in evaluation of machine translation systems. In *Proceedings of the 13th conference on Computational Linguistics*, volume 2, pages 211–216, Morristown, NJ, USA. Association for Computational Linguistics.

Sabine Lehmann, Stephan Oepen, Sylvie Regnier-Prost, Klaus Netter, Veronika Lux, Judith Klein, Kirsten Falkedal, Frederik Fouvry, Dominique Estival, Eva Dauphin, Herve Compagnion, Judith Baur, Lorna Balkan, and Doug Arnold. 1996. TSNLP - Test Suites for Natural Language Processing. *Proceedings of the 16th ...*, page 7.

| category | phenomena |
|---|---|
| Ambiguity | lexical ambiguity, structural ambiguity |
| Composition | phrasal verb, compound |
| Coordination & ellipsis | slicing, right-node rasing, gapping, stripping |
| False friends | |
| Function word | focus particle, modal particle, question tag |
| Long-distance dependency (LDD) & interrogative | multiple connectors, topicalization, polar question, WH-movement, scrambling, extended adjective construction, extraposition, pied-piping |
| Multi-word expression | prepositional MWE, verbal MWE, idiom, collocation |
| Named entity (NE) & terminology | date, measuring unit, location, proper name, domain-specific term |
| Negation | |
| Non-verbal agreement | coreference, internal possessor, external possessor |
| Punctuation | comma, quotation marks |
| Subordination | adverbial clause, indirect speech, cleft sentence, infinitive clause, relative clause, free relative clause, subject clause, object clause |
| Verb tense/aspect | future I, future II, perfect, pluperfect, present, preterite, progressive |
| mood | indicative, imperative, subjunctive, conditional |
| type | ditransitive, transitive, intransitive, modal, reflexive |
| Verb valency | case government, passive voice, mediopassive voice, resultative predicates |

Table 1: Categorization of the grammatical phenomena

Vivien Macketanz, Renlong Ai, Aljoscha Burchardt, and Hans Uszkoreit. 2018. TQ-AutoTest An Automated Test Suite for (Machine) Translation Quality. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation. International Conference on Language Resources and Evaluation (LREC-2018), 11th, May 7-12, Miyazaki, Japan*. European Language Resources Association (ELRA).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, John Makhoul, Ralph Weischedel, John Makhoul, and Ralph Weischedel. 2006. A Study of Translation Error Rate with Targeted Human Annotation. In *Proceedings of the 7th biennial conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA, USA. International Association for Machine Translation.

Felix Stahlberg, Adrià de Gispert, Eva Hasler, and Bill Byrne. 2016. Neural Machine Translation by Minimising the Bayes-risk with Respect to Syntactic Translation Lattices. *CoRR*, abs/1612.03791.

Andrew Way. 1991. Developer-Oriented Evaluation of MT Systems. In *Proceedings of the Evaluators' Forum*, pages 237–244, Les Rasses, Vaud, Switzerland. ISSCO.

| | # | JHU | LMU | MLLP | NJUNMT | NTT | onl-A | onl-B | onl-F | onl-G | onl-Y | RWTH | Ubiqus | UCAM | uedin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ambiguity | 76 | **69.7** | **64.5** | 55.3 | 59.2 | **63.2** | **68.4** | **73.7** | 42.1 | **71.1** | **65.8** | **78.9** | **64.5** | **76.3** | 51.3 |
| False friends | 34 | **61.8** | **58.8** | **61.8** | 50.0 | **73.5** | **76.5** | **79.4** | **70.6** | **76.5** | **67.6** | **70.6** | **55.9** | **67.6** | **55.9** |
| Verb valency | 30 | 80.0 | 73.3 | 86.7 | 83.3 | 86.7 | 86.7 | 86.7 | 70.0 | 76.7 | 86.7 | 90.0 | 86.7 | 86.7 | 86.7 |
| Verb tense/aspect/mood | 4110 | 74.3 | 65.8 | 84.4 | 61.6 | 84.3 | 71.8 | 72.1 | 75.2 | 45.8 | 70.6 | 78.3 | 73.5 | **86.9** | 76.7 |
| Non-verbal agreement | 48 | **75.0** | 60.4 | **79.2** | **77.1** | **83.3** | **75.0** | **87.5** | 50.0 | 60.4 | **81.3** | **85.4** | **75.0** | **81.3** | **75.0** |
| Punctuation | 51 | 60.8 | 56.9 | 62.7 | 56.9 | 68.6 | **96.1** | 74.5 | 3.9 | 60.8 | 76.5 | 58.8 | 82.4 | 52.9 | 66.7 |
| Subordination | 34 | **94.3** | **94.3** | **91.4** | **91.4** | **94.3** | **94.3** | **85.7** | 45.7 | **77.1** | **94.3** | **88.6** | **88.6** | **91.4** | **91.4** |
| MWE | 54 | **63.0** | **55.6** | **59.3** | **66.7** | **66.7** | **68.5** | **75.9** | 42.6 | **70.4** | **75.9** | **70.4** | **61.1** | **66.7** | **61.1** |
| LDD & interrogatives | 40 | **80.0** | **77.5** | **80.0** | **88.0** | **82.5** | **82.5** | **85.0** | 60.0 | **77.5** | **85.0** | **87.5** | **82.5** | **87.5** | **75.0** |
| NE & terminology | 35 | 82.9 | 80.0 | 88.6 | 80.0 | 88.6 | 77.1 | 77.1 | 77.1 | 77.1 | 91.4 | 82.9 | 77.1 | 80.0 | 80.0 |
| Coordination & ellipsis | 24 | **79.2** | **79.2** | **87.5** | **79.2** | **87.5** | **87.5** | **87.5** | 25.0 | **58.3** | **87.5** | **87.5** | **87.5** | **87.5** | **83.3** |
| Negation | 20 | **100.0** | **90.0** | **100.0** | **95.0** | **100.0** | **100.0** | **95.0** | 65.0 | 60.0 | **100.0** | **100.0** | **95.0** | **100.0** | **100.0** |
| Composition | 43 | **83.7** | 60.5 | **81.4** | 69.8 | **88.4** | **79.1** | **95.3** | **76.7** | 72.1 | **88.4** | **88.4** | **76.7** | **93.0** | **76.7** |
| Function word | 50 | **72.0** | 56.0 | **76.0** | 52.0 | **76.0** | **82.0** | **76.0** | 38.0 | 48.0 | **86.0** | **88.0** | **78.0** | **80.0** | **78.0** |
| Sum | 4650 | 3463 | 3071 | 3873 | 2913 | 3893 | 3393 | 3413 | 3379 | 2262 | 3344 | 3663 | 3438 | 4005 | 3547 |
| Non-weighted average | | 74.4 | 66.0 | 83.3 | 62.6 | 83.7 | 72.9 | 73.3 | 72.6 | 48.5 | 71.9 | 78.7 | 73.8 | **86.0** | 76.2 |
| Weighted average | | 76.9 | 69.5 | 78.2 | 71.6 | 81.7 | 81.8 | 82.2 | 53.0 | 66.6 | 82.6 | 82.5 | 77.5 | 81.3 | 75.6 |

Table 2: System accuracy (%) on each error category based on Analysis 1, having removed all test sentences whose evaluation remained uncertain, even for one of the systems. Boldface indicates the significantly best systems in the category

| | JHU | LMU | LMU-uns | MLLP | NJUNMT | NTT | onl-A | onl-B | onl-F | onl-G | onl-Y | RWTH-uns | RWTH | Ubiqus | UCAM | uedin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ambiguity | 65.4 | 61.3 | 26.9 | 53.1 | 59.5 | 59.3 | 66.7 | 74.7 | 43.2 | 70.4 | 67.1 | 29.1 | 80.3 | 63.6 | 72.4 | 48.2 |
| False friends | 58.3 | 55.6 | 50.0 | 58.3 | 47.2 | 73.5 | 72.2 | 75.0 | 66.7 | 72.2 | 66.7 | 41.7 | 66.7 | 52.8 | 63.9 | 52.8 |
| Verb valency | 76.7 | 67.2 | 60.0 | 84.5 | 76.8 | 86.4 | 81.4 | 84.1 | 55.3 | 67.9 | 89.5 | 50.0 | 91.7 | 78.7 | 80.7 | 74.2 |
| Verb tense/aspect/mood | 73.4 | 64.6 | 29.4 | 83.0 | 61.0 | 83.7 | 71.5 | 71.9 | 74.3 | 44.9 | 70.4 | 42.5 | 77.6 | 72.2 | 85.7 | 75.8 |
| Non-verbal agreement | 77.4 | 56.1 | 38.3 | 81.5 | 75.9 | 85.5 | 77.6 | 89.7 | 46.4 | 59.7 | 79.3 | 39.6 | 86.0 | 72.9 | 82.5 | 75.0 |
| Punctuation | 63.8 | 59.6 | 50.0 | 65.5 | 60.7 | 70.2 | 96.5 | 75.4 | 3.5 | 61.1 | 76.8 | 51.0 | 61.4 | 84.2 | 56.9 | 69.0 |
| Subordination | 98.8 | 98.5 | 79.2 | 96.4 | 97.6 | 97.8 | 98.8 | 95.7 | 70.5 | 89.7 | 98.9 | 75.0 | 96.7 | 96.3 | 96.4 | 97.3 |
| MWE | 63.1 | 49.3 | 11.1 | 56.3 | 62.7 | 62.9 | 63.8 | 70.8 | 40.9 | 65.7 | 66.2 | 17.4 | 67.1 | 58.6 | 64.8 | 57.4 |
| LDD & interrogatives | 83.7 | 72.6 | 73.3 | 86.4 | 81.8 | 86.8 | 83.1 | 88.3 | 53.6 | 56.2 | 86.0 | 81.3 | 93.3 | 85.4 | 88.7 | 76.6 |
| NE & terminology | 86.5 | 79.7 | 84.4 | 86.5 | 83.3 | 87.7 | 78.4 | 81.8 | 75.0 | 72.6 | 84.3 | 91.7 | 87.0 | 77.8 | 78.9 | 79.5 |
| Coordination & ellipsis | 83.1 | 80.0 | 61.5 | 84.1 | 78.0 | 88.9 | 86.7 | 92.9 | 26.1 | 55.1 | 80.0 | 18.5 | 88.0 | 84.1 | 90.7 | 86.5 |
| Negation | 100.0 | 90.0 | 23.5 | 100.0 | 95.0 | 100.0 | 100.0 | 95.0 | 65.0 | 60.0 | 100.0 | 57.1 | 100.0 | 95.0 | 100.0 | 100.0 |
| Composition | 81.3 | 59.2 | 60.0 | 79.2 | 67.3 | 85.4 | 77.6 | 93.6 | 76.1 | 70.8 | 87.5 | 14.6 | 85.7 | 73.5 | 89.8 | 73.5 |
| Function word | 70.4 | 51.4 | 15.7 | 78.3 | 50.7 | 76.4 | 81.2 | 75.3 | 32.8 | 47.1 | 84.7 | 21.7 | 81.9 | 71.2 | 80.6 | 75.3 |
| Sum | 5328 | 5287 | 1795 | 5303 | 5231 | 5331 | 5312 | 5357 | 5211 | 5171 | 5309 | 2341 | 5362 | 5296 | 5351 | 5300 |
| Non-weighted average | 74.1 | 65.0 | 31.8 | 82.2 | 62.9 | 83.3 | 73.1 | 73.9 | 70.6 | 47.9 | 72.3 | 41.6 | 78.8 | 73.0 | 84.9 | 75.5 |
| Weighted average | 77.3 | 67.5 | 47.4 | 78.1 | 71.3 | 81.7 | 81.1 | 83.2 | 52.1 | 63.8 | 81.2 | 45.1 | 83.1 | 76.2 | 80.9 | 74.4 |

Table 3: System accuracy (%) on each error category based on Analysis 2, having removed only the system outputs whose evaluation remained uncertain.

|  | # | JHU | LMU | MLLP | NJUNMT | NTT | onl-A | onl-B | onl-F | onl-G | onl-Y | RWTH | Ubiqus | UCAM | uedin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Future I | 494 | **70.2** | 68.3 | **70.3** | 56.8 | **69.7** | 64.8 | 65.7 | 64.8 | 45.8 | 63.8 | 68.8 | 68.5 | **73.8** | **69.3** |
| Future I subjunctive II | 479 | 66.8 | 56.3 | **75.7** | 48.3 | **74.5** | 63.8 | 60.5 | 44.2 | 39.8 | 59.3 | 64.7 | 63.7 | **73.8** | 66.7 |
| Future II | 138 | **28.8** | **27.5** | **28.5** | **23.8** | **27.8** | 21.0 | **31.0** | **25.3** | 12.0 | **25.8** | **30.3** | **28.5** | **30.8** | **29.0** |
| Future II subjunctive II | 128 | **27.5** | 17.0 | **27.5** | **23.3** | **29.5** | **29.3** | **31.0** | **26.3** | 18.0 | 30.0 | **26.5** | 11.3 | **30.3** | **25.3** |
| Perfect | 506 | 64.8 | 51.2 | **80.5** | 60.0 | **79.7** | 57.0 | 65.5 | 67.7 | 26.5 | 67.7 | 74.8 | 62.8 | **78.0** | 70.3 |
| Pluperfect | 478 | 39.7 | 24.7 | **57.0** | 23.5 | **52.5** | 22.3 | 26.5 | **54.2** | 16.0 | 5.0 | 30.2 | 33.7 | **52.3** | 50.2 |
| Pluperfect subjunctive II | 442 | 37.2 | 35.8 | **53.0** | 37.3 | **54.2** | 41.0 | 40.7 | 50.3 | 13.7 | 49.5 | **52.2** | 42.7 | **56.3** | 41.3 |
| Present | 482 | 71.3 | 68.3 | **73.5** | 58.5 | 69.7 | 69.8 | 59.8 | 68.8 | 50.2 | 61.8 | **75.5** | 72.0 | **77.7** | **75.0** |
| Preterite | 513 | 69.3 | 66.2 | 74.3 | 59.3 | 79.5 | **81.2** | 78.0 | 70.7 | 60.2 | **80.7** | 76.2 | 76.3 | **83.7** | 64.3 |
| Preterite subjunctive II | 433 | 49.8 | 48.0 | **54.2** | 44.2 | **57.2** | **56.0** | **53.5** | **58.3** | 39.3 | **56.0** | **53.5** | **55.0** | **56.2** | 49.3 |
| Sum/non-weighted average | 4093 | 54.3 | 48.1 | 61.7 | 44.9 | 61.6 | 52.4 | 52.7 | 55.0 | 33.4 | 51.5 | 57.2 | 53.7 | 63.5 | 56.0 |
| Weighted average |  | 52.7 | 46.5 | 59.7 | 43.6 | 59.6 | 50.8 | 51.4 | 53.2 | 32.3 | 50.1 | 55.4 | 51.7 | 61.5 | 54.2 |

Table 4: System accuracy (%) on linguistic phenomena related to verb tenses

|  | # | JHU | LMU | MLLP | NJUNMT | NTT | onl-A | onl-B | onl-F | onl-G | onl-Y | RWTH | Ubiqus | UCAM | uedin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ditransitive | 329 | 85.7 | 82.7 | 90.0 | 70.8 | 93.6 | 77.8 | 88.8 | 79.0 | 61.1 | 81.5 | 90.0 | 79.0 | **95.4** | 83.9 |
| Intransitive | 397 | 77.1 | 77.6 | 89.2 | 64.5 | 84.6 | 83.9 | 89.9 | 78.1 | 69.3 | 87.7 | 90.4 | 84.1 | **93.7** | 86.6 |
| Modal | 1353 | 69.0 | 56.4 | **82.6** | 56.2 | **81.5** | 65.9 | 68.1 | 73.7 | 39.2 | 65.0 | 73.1 | 69.3 | **81.7** | 71.8 |
| Modal negated | 1403 | 73.5 | 65.6 | **85.3** | 59.2 | 82.2 | 66.6 | 59.0 | 74.8 | 36.4 | 63.2 | 74.1 | 71.7 | **86.2** | 75.2 |
| Reflexive | 246 | 74.0 | 50.8 | 58.9 | 45.1 | **80.9** | 75.6 | **85.4** | 64.6 | 30.9 | 73.2 | 69.9 | 61.8 | **80.1** | 71.5 |
| Transitive | 365 | 83.6 | 82.7 | 94.8 | 89.0 | **96.2** | 92.1 | 93.2 | 83.6 | 75.6 | 88.8 | 95.1 | 86.8 | **98.1** | 85.8 |
| Sum/non-weighted average | 4093 | 74.3 | 65.7 | 84.4 | 61.5 | 84.3 | 71.8 | 72.0 | 75.3 | 45.7 | 70.5 | 78.2 | 73.5 | 86.9 | 76.6 |
| Weighted average |  | 77.2 | 69.3 | 83.5 | 64.1 | 86.5 | 77.0 | 80.7 | 75.6 | 52.1 | 76.6 | 82.1 | 75.5 | 89.2 | 79.1 |

Table 5: System accuracy (%) on linguistic phenomena related to verb types

|  | # | JHU | LMU | MLLP | NJUNMT | NTT | onl-A | onl-B | onl-F | onl-G | onl-Y | RWTH | Ubiqus | UCAM | uedin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Compound | 26 | 73.1 | 73.1 | 73.1 | 69.2 | **84.6** | **80.8** | **92.3** | **76.9** | **80.8** | **84.6** | **80.8** | **76.9** | **88.5** | **76.9** |
| Quotation marks | 38 | 47.4 | 42.1 | 50.0 | 42.1 | 60.5 | **94.7** | 68.4 | 0.0 | 50.0 | 68.4 | 44.7 | 76.3 | 36.8 | 55.3 |
| Phrasal verb | 17 | **100.0** | 58.8 | **94.1** | 70.6 | **94.1** | 76.5 | **100.0** | 76.5 | 58.8 | **94.1** | **100.0** | 76.5 | **100.0** | 76.5 |
| Question tag | 15 | 66.7 | 20.0 | **86.7** | 0.0 | 73.3 | 66.7 | 60.0 | 0.0 | 13.3 | **93.3** | **100.0** | 73.3 | **86.7** | **80.0** |
| Collocation | 15 | 60.0 | 40.0 | 53.3 | 60.0 | 60.0 | **66.7** | **86.7** | 20.0 | 80.0 | **86.7** | 60.0 | 60.0 | **66.7** | 60.0 |
| Location | 15 | 93.3 | 86.7 | 93.3 | 86.7 | 100.0 | 86.7 | 86.7 | 93.3 | 93.3 | 100.0 | 93.3 | 93.3 | 93.3 | 86.7 |
| Modal particle | 16 | 56.3 | 50.0 | 56.3 | 50.0 | **75.0** | **93.8** | **81.3** | 18.8 | 50.0 | **87.5** | **75.0** | 56.3 | 62.5 | 56.3 |

Table 6: System accuracy (%) on specific linguistic phenomena with more than 15 test sentences

# The Word Sense Disambiguation Test Suite at WMT18

**Annette Rios**[1] **Mathias Müller**[1] **Rico Sennrich**[1,2]

[1]Institute of Computational Linguistics, University of Zurich
{rios,mmueller}@cl.uzh.ch

[2]School of Informatics, University of Edinburgh
rico.sennrich@ed.ac.uk

## Abstract

We present a task to measure an MT system's capability to translate ambiguous words with their correct sense according to the given context. The task is based on the German–English Word Sense Disambiguation (WSD) test set ContraWSD (Rios Gonzales et al., 2017), but it has been filtered to reduce noise, and the evaluation has been adapted to assess MT output directly rather than scoring existing translations. We evaluate all German–English submissions to the WMT'18 shared translation task, plus a number of submissions from previous years, and find that performance on the task has markedly improved compared to the 2016 WMT submissions (81%→93% accuracy on the WSD task). We also find that the unsupervised submissions to the task have a low WSD capability, and predominantly translate ambiguous source words with the same sense.

## 1 Introduction

Ambiguous words are often difficult to translate automatically, since the MT system has to decide which sense is correct in the given context. Errors in lexical choice can result in bad or even incomprehensible translations. However, document-level metrics, such as BLEU (Papineni et al., 2002) are not fine-grained enough to assess this type of error.

Early evaluations have shown that neural machine translation (NMT) produces translations that are substantially more *fluent*, i.e. more grammatical and natural, than the previously dominant phrase-based/syntax-based statistical models, but results are more mixed when comparing *adequacy*, the semantic faithfulness of the translation to the original (Bojar et al., 2016; Bentivogli et al., 2016; Castilho et al., 2017; Klubička et al., 2017).

For example, in the fine-grained human evaluation by Klubička et al. (2017), *mistranslations* were the most frequent error category for the NMT system they evaluated, whereas fluency errors dominated in phrase-based machine translation.[1] Our aim is to quantify one aspect of adequacy, word sense disambiguation (WSD), in a reproducible and semi-automatic way, to track progress over time and compare different types of systems in this respect.

We present a German→English test set to semi-automatically assess an MT systems performance on word sense disambiguation. The test set is based on ContraWSD (Rios Gonzales et al., 2017), but has been further filtered to reduce noise, and we use a different evaluation protocol. Instead of scoring a set of translations and measuring whether the reference translation is scored highest, we base the evaluation on the 1-best translation output to make the evaluation applicable to black-box systems. We report results on all German→English submissions to the WMT 2018 shared translation task (Bojar et al., 2018), plus a number of baseline systems from previous years.

## 2 Test Suite

Rather than measuring word sense disambiguation against a manually defined sense inventory such as those in Wordnet (Miller, 1995), we perform a task-based evaluation, focusing on homonyms whose different senses have distinct translations.[2]

[1] Note that, while mistranslations were the most frequent error category in NMT, their absolute number was still lower in the NMT system output than in the phrase-based one.

[2] Other task-based evaluation sets for word sense disambiguation include (Lefever and Hoste, 2013; Gorman et al., 2018).

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 588–596
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64064

The collection of test cases consists of 3249 German–English sentence pairs where the German source contains one of 20 ambiguous words that have more than one possible translation in English.[3] We have associated the 20 ambiguous words with a total of 45 word senses, and extracted up to 100 examples for each sense.

The set of ambiguous words and sentence pairs are based on the test set described in (Rios Gonzales et al., 2017).[4] The original test set was designed to use scoring for the evaluation, however, in the present task we let the systems translate the source sentences, and evaluate the translation output. This change in evaluation protocol required further filtering of the original test set, specifically, the removal of German words with an English translation that covers multiple senses. For instance, the original test set contains *Stelle* with two English senses: *job* and *place*. Both meanings can be translated as *position*, in which case we would not be able to assess the translation as correct or wrong, therefore *Stelle* was removed from our set of ambiguous words.

Since for most ambiguous words, one or more of their meanings are relatively rare, a large amount of parallel text is necessary to extract a sufficiently balanced number of examples.[5] The correct translation is automatically determined for each pair through the reference translation. Table 1 lists all the ambiguous German words in the test set with with their translations in English. We base our statistics on the number of ambiguous source words, which is slightly higher (3363) than the number of sentences (3249). Sentence pairs where the reference translation contains more than one possible sense as a translation have been removed. For instance, if a given reference contains the word *investment* as a translation for *Anlage*, but also *attachment* as a translation of another source word, this sentence pair cannot be part of the test set, since word alignment would be required to assess it correctly.

The evaluation is semi-automatic: We automatically check for each sentence in the MT output if one of the correct translations of the ambiguous word is present, and if the output contains one of the other possible translations of the word, i.e. if it has been translated with one of its other senses. Note that we check for more variation in the automatic matching than shown in Table 1, e.g. for *Absatz - sales*, we also consider verbal forms such as *sold, sells, selling etc.* as correct, using manually created lists of valid translations.[6]

There are four possible outcomes of this automatic evaluation:

1. we find only instances of the correct translations → counts as correct[7]
2. we find only instances of the other translations → counts as wrong
3. we find both the correct and one of the other translations → manual inspection
4. we find none of the known translations → manual inspection

## 2.1 Manual Evaluation Protocol

The large majority of translation outputs could be categorized as correct or wrong automatically. For the remaining approximately 5%, we manually assigned a label. Overall, around 25% of these were labelled as correct.

Case 3 typically indicates that the same ambiguous source word occurs multiple times in the input, and a manual annotator provided the number

[3] The test set and evaluation scripts are available from https://github.com/a-rios/ContraWSD/tree/master/testsuite_wmt18

[4] The identification of ambiguous words and senses was performed with the help of lexical translation probabilities.

[5] Sentence pairs have been extracted from the following corpora:

- WMT test and development sets 2006-2016 (de-en) and 2006-2013 (de-fr)
- Crédit Suisse News Corpus https://pub.cl.uzh.ch/projects/b4c/de/
- Corpora from OPUS (Tiedemann, 2012):
  - Global Voices (http://opus.lingfil.uu.se/GlobalVoices.php)
  - Books (http://opus.lingfil.uu.se/Books.php)
  - EU Bookshop Corpus (http://opus.lingfil.uu.se/EUbookshop.php)
- MultiUN (Ziemski et al., 2016)

[6] Since we do not use word alignment, there is a risk that we mistakenly match a translation from another part of the sentence. However, this is only a problem in the rare case where, at the same time, the ambiguous source word itself is not translated into a known translation, since conflicting matches trigger a manual inspection.

[7] If there are multiple instances of the ambiguous source word in the sentence, we automatically count the number of correct translations to assign credit.

| word | translations/senses | | | |
|---|---|---|---|---|
| | sense 1 | sense 2 | sense 3 | sense 4 |
| Absatz | heel | paragraph | sales | |
| Anlage | attachment, annex | installation, facility, plant | investment | |
| Annahme | acceptance, approval | assumption, conjecture | | |
| Aufgabe | abandonment, surrender | task, exercise | | |
| Auflösung | dissolution, liquidation | resolution | | |
| Decke | blanket, cover | ceiling | | |
| Einsatz | bet | commitment | usage, application | |
| Gericht | court, tribunal | dish, meal | | |
| Himmel | heaven | sky | | |
| Karte | card | menu | ticket | map |
| Kurs | course, class | price, rate | | |
| Lager | storage, stock | camp | | |
| Opfer | victim | sacrifice | | |
| Preis | prize | price, cost, fee | | |
| Rat | advice, counsel | council, board | | |
| Raum | region,area | room, space | | |
| Schlange | serpent, snake | queue, line | | |
| Ton | tone, sound | clay | | |
| Tor | door, gate | goal | | |
| Wahl | election | choice, selection | | |

Table 1: List of ambiguous German words, and the English translations of their different senses, included in the test suite.

| | |
|---|---|
| source | Im Allgemeinen lässt sich deshalb mit Recht behaupten, dass – mit der richtigen Beratung und Sorgfalt – Hedge-Fund-**Anlagen** nicht zwangsläufig risikoreicher sind als traditionelle **Anlagen**. |
| reference | It is therefore fair to say that properly advised hedge fund **investments** are, generally speaking, not necessarily riskier than traditional **investments**. |
| MT translation | In general, therefore, it is fair to say that, with the right advice and care, hedge fund **assets** are not necessarily more risky than traditional **plants**. |

Table 2: Example sentence pair for ambiguous word *Anlagen* with translation from uedin-nmt-2017. The first translation *assets* is correct, the second (*plants*) wrong.

of correct translations. See Table 2 with an example from one of the baseline systems, where the ambiguous word *Anlage* occurs twice, both times in the financial sense. The MT system translates the first form correctly, but the second with one of its other meanings, *plant*.

Case 4 can indicate that the ambiguous source word was translated into a variant not covered by our automatic patterns, or left untranslated.[8] Manual assessment by the main author is used to distinguish between the two.

## 3 Evaluation

We present results for all submissions to the WMT'18 shared translation task for German→English. In addition, we include several baseline systems in our evaluation to track performance over time. We report results for Edinburgh's WMT'16 and WMT'17 submitted neural systems for German→English (Sennrich et al., 2016, 2017), which were ranked first in 2016, and tied first in 2017.[9] We also include Edinburgh's WMT'16 syntax-based system (Williams et al., 2016), ranked tied second in 2016, to compare the now dominant neural systems to a more traditional SMT system.

We report the WSD accuracy for each system, in two variants: automatic and full. For automatic accuracy only case 1 is considered correct, and cases 2–4 are considered wrong. Full accuracy considers some cases 3 and 4 (where both a correct and an incorrect translation, or none of the listed translations, are found) correct, if they were found to be correct upon manual inspection. We also report BLEU scores on newstest2018, and on the WSD test suite, for comparison.

## 4 Results

Results on the WSD test suite are shown in Table 3. Table 4 shows an error analysis with two categories, distinguishing between predicting the wrong sense, and leaving the ambiguous source word untranslated. Globally, we observe a strong correlation between WSD accuracy and BLEU on the WSD test suite (Kendall's $\tau = 0.91$), and a smaller (but still strong) correlation between WSD accuracy and BLEU on newstest2018 ($\tau = 0.72$). However, there are some notable differences between BLEU and WSD accuracy. Especially some unconstrained, anonymous systems (online-A/B/G/Y) perform better on the WSD test suite than newstest2018 relative to other systems, which is likely due to differences in domain focus and training data: most constrained systems built for the shared task use monolingual news data for domain adaptation, whereas the online systems likely do not. At the same time, the online systems may be using extra training resources, and we cannot rule out that they train on corpora from which the WSD test suite is extracted.

The unsupervised systems RWTH-UNSUPER and LMU-unsup, as well as the anonymous rule-based system online-F clearly fall behind. In many cases, these systems stick to one translation of a given ambiguous word. This becomes obvious when looking at the number of cases where the translation contains one of the other meanings of the translated words. The less common a given sense, the more likely it is translated with one of its other meanings - this is true for all systems, but more pronounced in the unsupervised models. Not only do they translate words with a wrong meaning more often, they seem to have learned some spurious correlations. For instance, the German word *Preis* (*price/prize*) was translated in almost all cases as *call* by LMU-unsup. Generally, the unsupervised systems tend to translate words in a deterministic fashion, i.e. they use mostly the same translation for an ambiguous source word, regardless of context.

We observe that there is little difference in WSD accuracy between the syntax-based and neural uedin systems from 2016, even though the neural system achieves a substantially higher BLEU score. This is consistent with human comparisons of statistical and neural systems at the time, which found large improvements in fluency, but only small differences in adequacy, or specifically the number of mistranslations (Bojar et al., 2016; Castilho et al., 2017; Klubička et al., 2017). Interestingly, we observe major improvements in lexical choice since the 2016 systems, with a jump of 5 percentage points in 2017, and another 8 percentage points by the best system in 2018.

While these experiments were not under controlled data conditions[10], we believe that this im-

[8] This includes cases where the original source word is used in the translation.

[9] Available at `http://data.statmt.org/wmt16_systems/` and `http://data.statmt.org/wmt17_systems/`

[10] There was a moderate improvement in the amount of training data in 2017 through the inclusion of the Rapid corpus of EU press releases (+20%), and a large increase in 2018

| system | WSD accuracy | | BLEU | |
|---|---|---|---|---|
| | automatic | full | newstest2018 | WSD test suite |
| uedin-syntax-2016 | 79.7 | 81.3 | 36.1 | 26.9 |
| uedin-nmt-2016 | 79.8 | 81.1 | 41.3 | 27.7 |
| uedin-nmt-2017 | 84.9 | 86.3 | 43.5 | 30.5 |
| RWTH | 92.4 | 93.6 | 48.4 | 33.6 |
| UCAM | 91.1 | 92.4 | 48.0 | 32.9 |
| online-B | 89.4 | 91.3 | 43.9 | 32.5 |
| NTT | 89.7 | 91.2 | 46.8 | 32.6 |
| JHU | 88.9 | 90.3 | 45.3 | 31.7 |
| online-Y | 88.0 | 89.8 | 39.5 | 30.9 |
| MLLP-UPV | 88.4 | 89.7 | 45.1 | 30.7 |
| uedin | 87.1 | 88.8 | 43.9 | 30.8 |
| Ubiqus-NMT | 86.7 | 88.3 | 44.1 | 31.0 |
| online-A | 86.6 | 88.0 | 40.6 | 29.7 |
| online-G | 85.4 | 86.9 | 31.9 | 29.1 |
| NJUNMT-private | 84.3 | 86.0 | 38.3 | 28.2 |
| LMU-nmt | 80.4 | 81.7 | 40.9 | 28.1 |
| online-F | 50.7 | 51.4 | 22.0 | 15.8 |
| RWTH-UNSUPER | 44.9 | 47.2 | 18.6 | 11.4 |
| LMU-unsup | 42.6 | 43.3 | 17.9 | 10.0 |

Table 3: Results on WSD test suite. WSD accuracy before and after manual inspection, and BLEU on newstest2018, and on references from WSD test suite.

| system | wrong sense | untranslated |
|---|---|---|
| uedin-syntax-2016 | 17.4 | 1.3 |
| uedin-nmt-2016 | 16.5 | 2.4 |
| uedin-nmt-2017 | 11.7 | 2.1 |
| RWTH | 5.2 | 1.2 |
| UCAM | 6.4 | 1.2 |
| online-B | 6.5 | 2.1 |
| NTT | 7.0 | 1.8 |
| JHU | 8.5 | 1.2 |
| online-Y | 9.0 | 1.2 |
| MLLP-UPV | 9.5 | 0.8 |
| uedin | 10.1 | 1.2 |
| Ubiqus-NMT | 9.3 | 2.3 |
| online-A | 11.0 | 1.0 |
| online-G | 11.6 | 1.5 |
| NJUNMT-private | 9.3 | 4.7 |
| LMU-nmt | 16.3 | 2.1 |
| online-F | 47.8 | 0.7 |
| RWTH-UNSUPER | 48.9 | 3.9 |
| LMU-unsup | 49.8 | 6.9 |

Table 4: Proportion of ambiguous words translated with the wrong sense, or left untranslated (in %).

provement is only partially explainable by the increase in the amount of training data. We highlight a number of systems to illustrate this point.

Paracrawl is a noisy resource, and most submission systems report using a filtered version of it. Ubiqus-NMT does not use Paracrawl at all, and is thus comparable to uedin-nmt-2017 in terms of training data, but outperforms it in WSD accuracy. This is even more impressive considering that Ubiqus-NMT is based on a single model, outperforming the reranked ensembles of uedin-nmt-2017.

A second interesting comparison is that between different architectures. LMU-nmt is based on a shallow RNN encoder-decoder, similar to uedin-nmt-2016, and exhibits a similarly low WSD accuracy. Most submissions are based on deep Transformer or RNN architectures, and show a higher WSD accuracy. Neural network depth was also one of the main differences between uedin-nmt-2016 and uedin-nmt-2017, and our results indicate that this is an important factor for lexical choice. Experiments by Tang et al. (2018), conducted in parallel to this work, on WMT17 training data also show that neural architectures

through the inclusion of Paracrawl (+700%).

play an important role in the performance on WSD, with a substantial lead for the Transformer over the tested RNN and CNN architectures.

The error analysis in Table 4 exposes other differences between systems. The rule-based system online-F is least prone to leaving the ambiguous source words untranslated (0.7%), while this is a more serious problems in the unsupervised systems (up to 6.9%) and some neural systems (up to 4.7%). It has been argued that SMT, which uses a coverage mechanism during decoding, is less prone to undertranslation than NMT (Tu et al., 2016). On the WSD test set, we find that uedin-nmt-2016 leaves more of the ambiguous words untranslated (2.4%) than the contemporaneous uedin-syntax-2016 (1.3%), but most NMT systems submitted to this year's shared translation task improve upon this number. While this is a very narrow evaluation of the undertranslation problem (only on one data set, and looking at specific source words), we consider it encouraging that we could measure some progress.

## 5 Conclusions

We present a targeted evaluation of 16 systems regarding their performance in lexical choice. A comparison against a baseline consisting of the top ranked systems from WMT 2016 and 2017 for German-English shows that translation models in general have improved substantially. Furthermore, we observe that unsupervised systems are at a clear disadvantage when it comes to word sense disambiguation: they are less flexible and tend to stick to one translation of a given ambiguous word, regardless of context.

The current study is focused on a small set of 20 ambiguous nouns and 45 word senses, and a large-scale test set is created by extracting 3249 sentence pairs containing one of these word senses from various parallel corpora. This focus on ambiguous source words without lexical overlap between word senses in the target language allowed us to define an evaluation protocol that is mostly automatic: manual inspection was only necessary for about $\approx 5\%$ of sentences, and had little effect on the ranking. However, this narrow focus also comes with limitations, and it would be interesting to evaluate word sense disambiguation on a larger set of words, and including other parts-of-speech such as verbs and adverbs, which constituted a substantial proportion of lexical choice errors in previous analyses of MT systems (Williams et al., 2015).

## Acknowledgement

We are grateful to the Swiss National Science Foundation (SNF) for support of the project CoNTra (grant number 105212_169888). We thank the anonymous reviewers for their helpful feedback.

## References

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 257–267, Austin, Texas. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation (WMT16). In *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Rico Sennrich, Vilelmini Sosoni, Yota Georgakopoulou, Pintu Lohar, Andy Way, Antonio Valerio Miceli Barone, and Maria Gialama. 2017. A Comparative Quality Evaluation of PBSMT and NMT using Professional Translators. In *Proceedings of Machine Translation Summit XVI*, Nagoya, Japan.

Kyle Gorman, Gleb Mazovetskiy, and Vitaly Nikolaev. 2018. Improving homograph disambiguation with supervised machine learning. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Filip Klubička, Antonio Toral, and Víctor M. Sánchez-Cartagena. 2017. Fine-Grained Human Evaluation of Neural Versus Phrase-Based Machine Translation. *The Prague Bulletin of Mathematical Linguistics*, 108:121–132.

Els Lefever and Véronique Hoste. 2013. SemEval-2013 Task 10: Cross-lingual Word Sense Disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 158–166, Atlanta, Georgia, USA. Association for Computational Linguistics.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. 2017. Improving Word Sense Disambiguation in Neural Machine Translation with Sense Embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's Neural MT Systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures. In *EMNLP 2018*, Brussels, Belgium. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling Coverage for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, Barry Haddow, and Ondřej Bojar. 2016. Edinburgh's Statistical Machine Translation Systems for WMT16. In *Proceedings of the First Conference on Machine Translation*, pages 399–410, Berlin, Germany. Association for Computational Linguistics.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, and Philipp Koehn. 2015. Edinburgh's Syntax-Based Systems at WMT 2015. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 199–209, Lisbon, Portugal. Association for Computational Linguistics.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations Parallel Corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

## Appendix A

| system | WSD accuracy | |
|---|---|---|
| | automatic | full |
| uedin-syntax-2016 | 77.6 | 79.3 |
| uedin-nmt-2016 | 77.7 | 79.1 |
| uedin-nmt-2017 | 83.0 | 84.6 |
| RWTH | 91.8 | 93.2 |
| UCAM | 90.3 | 91.7 |
| online-B | 88.5 | 90.6 |
| NTT | 88.5 | 90.3 |
| JHU | 87.7 | 89.3 |
| online-Y | 87.1 | 89.1 |
| MLLP-UPV | 87.2 | 88.7 |
| uedin | 85.6 | 87.5 |
| Ubiqus-NMT | 85.3 | 87.2 |
| online-A | 85.4 | 86.9 |
| online-G | 84.4 | 86.1 |
| NJUNMT-private | 83.1 | 85.1 |
| LMU-nmt | 78.1 | 79.6 |
| online-F | 48.3 | 49.0 |
| RWTH-UNSUPER | 38.5 | 41.2 |
| LMU-unsup | 38.3 | 38.9 |

Table 5: Results on WSD test suite, ignoring sentences from WMT dev/test data. WSD accuracy before and after manual inspection.

# LIUM-CVC Submissions for WMT18 Multimodal Translation Task

**Ozan Caglayan, Adrien Bardet,**
**Fethi Bougares, Loïc Barrault**
LIUM, Le Mans University
FirstName.LastName@univ-lemans.fr

**Kai Wang, Marc Masana, Luis Herranz and Joost van de Weijer**
CVC, Universitat Autonoma de Barcelona
{kwang,mmasana,lherranz,joost}@cvc.uab.es

## Abstract

This paper describes the multimodal Neural Machine Translation systems developed by LIUM and CVC for WMT18 Shared Task on Multimodal Translation. This year we propose several modifications to our previous multimodal attention architecture in order to better integrate convolutional features and refine them using encoder-side information. Our final constrained submissions ranked first for English→French and second for English→German language pairs among the constrained submissions according to the automatic evaluation metric METEOR.

## 1 Introduction

In this paper, we present the neural machine translation (NMT) and multimodal NMT (MMT) systems developed by LIUM and CVC for the third edition of the shared task. Several lines of work have been conducted since the introduction of the shared task on MMT in 2016 (Specia et al., 2016). The majority of last year submissions including ours (Caglayan et al., 2017a) were based on the integration of global visual features into various parts of the NMT architecture (Elliott et al., 2017). Apart from these, hierarchical multimodal attention (Helcl and Libovický, 2017) and multi-task learning (Elliott and Kádár, 2017) were also explored by the participants.

This year we decided to revisit the multimodal attention (Caglayan et al., 2016) since our previous observations about qualitative analysis of the visual attention was not satisfying. In order to improve the multimodal attention both qualitatively and quantitatively, we experiment with several refinements to it: first, we try to use different input image sizes prior to feature extraction and second we normalize the final convolutional feature maps to assess its impact on the final MMT performance. In terms of architecture, we propose to refine the visual features by learning an encoder-guided early spatial attention. In overall, we find that normalizing feature maps is crucial for the multimodal attention to obtain a comparable performance to monomodal NMT while the impact of the input image size remains unclear. Finally, with the help of the refined attention, we obtain modest improvements in terms of BLEU (Papineni et al., 2002) and METEOR (Lavie and Agarwal, 2007).

The paper is organized as follows: data preprocessing, model details and training hyperparameters are detailed respectively in section 2 and section 3. The results based on automatic evaluation metrics are reported in section 4. Finally the paper ends with a conclusion in section 5.

## 2 Data

We use Multi30k (Elliott et al., 2016) dataset provided by the organizers which contains 29000, 1014, 1000 and 1000 English→{German,French} sentence pairs respectively for `train`, `dev`, `test2016` and `test2017`. A new training split of 30014 pairs is formed by concatenating the `train` and `dev` splits. Early-stopping is performed based on METEOR computed over the `test2016` set and the final model selection is done over `test2017`.

Punctuation normalization, lowercasing and aggressive hyphen splitting were applied to all sentences prior to training. A Byte Pair Encoding (BPE) model (Sennrich et al., 2016) with 10K merge operations is jointly learned on English-German and English-French resulting in vocabularies of 5189-7090 and 5830-6608 subwords respectively.

### 2.1 Visual Features

Since Multi30k images involve much more complex region-level relationships and scene compositions compared to ImageNet (Russakovsky et al.,

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 597–602
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64065

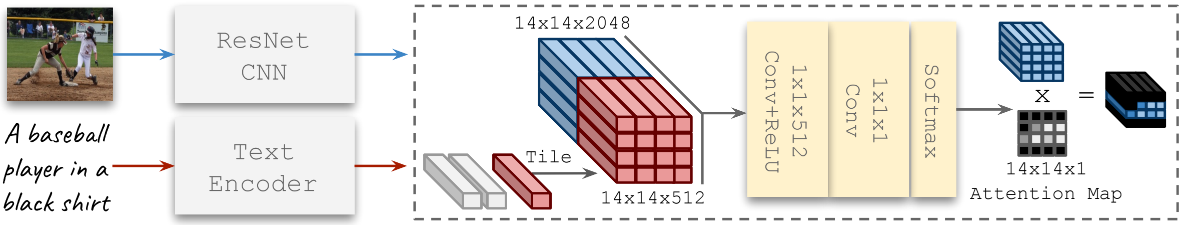


Figure 1: Filtered attention (FA): the convolutional feature maps are dynamically masked using an attention conditioned on the source sentence representation.

2015) object classification task, we explore different input image sizes to quantify its impact in the context of MMT since rescaling the input image has a direct effect on the size of the receptive fields of the CNN. After normalizing the images using ImageNet mean and standard deviation, we resize and crop the images to 224x224 and 448x448. Features are then extracted from the final convolutional layer (`res5c_relu`) of a pre-trained ResNet50 (He et al., 2016) CNN.[1] This led to feature maps $\mathrm{V} \in \mathbb{R}^{2048 \times w \times w}$ where the spatial dimensionality $w$ is 7 or 14.

#### 2.1.1 Feature Normalization

We conjecture that transferring ReLU features from a CNN into a model that only makes use of bounded non-linearities like *sigmoid* and *tanh*, can saturate the non-linear neurons in the very early stages of training if their weights are not carefully initialized. Instead of tuning the initialization, we experiment with $\mathrm{L}_2$ normalization over the channel dimension so that each feature vector ($\in \mathbb{R}^{2048}$) has an $\mathrm{L}_2$ norm of 1.

## 3 Models

In this section we will describe our baseline NMT and multimodal NMT systems. All models use 128 dimensional embeddings and GRU (Cho et al., 2014) layers with 256 hidden states. Dropout (Srivastava et al., 2014) is applied over source embeddings $x_s$, encoder states $\mathrm{H}^{enc}$ and pre-softmax activations $o_t$. We also apply $\mathrm{L}_2$ regularization with a factor of $1e{-}5$ on all parameters except biases. The parameters are initialized using the method proposed by He et al. (2015) and optimized with Adam (Kingma and Ba, 2014). The total gradient norm is clipped to 1 (Pascanu et al., 2013). We use batches of size 64 and an initial learning rate of $4e{-}4$. All systems are implemented using the PyTorch version of *nmtpy*[2] (Caglayan et al., 2017b).

### 3.1 Baseline NMT

Let us denote the length of the source sentence $\{x_1, \dots, x_S\}$ and the target sentence $\{y_1, \dots, y_T\}$ by $S$ and $T$ respectively. The source sentence is first encoded with a 2-layer bidirectional GRU to obtain the set of hidden states:

$$\mathrm{H}^{enc} \leftarrow Enc(\{x_1, \dots, x_S\}), \mathrm{H}^{enc} \in \mathbb{R}^{S \times 512}$$

The decoder is a 2-layer conditional GRU (CGRU) (Sennrich et al., 2017) with tied embeddings (Press and Wolf, 2016). CGRU is a stacked 2-layer recurrence block with the attention mechanism in the middle. We use feed-forward attention (Bahdanau et al., 2014) which encapsulates a learnable layer. The first decoder (which is initialized with a zero vector) receives the previous target embeddings as inputs (equation 1). At each timestep of the decoding stage, the attention mechanisms produces a context vector $c_t^{txt}$ (equation 2) that becomes the input to the second GRU (equation 3). Finally, the probability over the target vocabulary is conditioned over a transformation of the final hidden state $h_t^{dec_2}$ (equation 4, 5).

$$\begin{aligned}
h_t^{dec_1} &= \mathrm{DEC}_1(y_{t-1}, h_{t-1}^{dec_2}) && (1)\\
c_t^{txt} &= \mathrm{ATT}_{txt}(\mathrm{H}^{enc}, h_t^{dec_1}) && (2)\\
h_t^{dec_2} &= \mathrm{DEC}_2(c_t^{txt}, h_t^{dec_1}) && (3)\\
o_t &= \tanh(\mathbf{W_o} h_t^{dec_2} + b_o) && (4)\\
P(y_t) &= softmax(\mathbf{W_v} o_t) && (5)
\end{aligned}$$

### 3.2 Multimodal Attention (MA)

Our baseline multimodal attention (MA) system (Caglayan et al., 2016) applies a spatial attention mechanism (Xu et al., 2015) over the visual features. At each timestep $t$ of the decoding stage,

[1] We use torchvision for feature extraction.

[2] github.com/lium-lst/nmtpytorch

a multimodal context vector $c_t$ is computed and given as input to the second decoder (equation 3):

$$c_t = \mathbf{W_f}\left[c_t^{txt}; \mathbf{W_{vis}} c_t^{vis}\right] \tag{6}$$

$$c_t^{vis} = \mathrm{ATT}_{vis}(\mathrm{V}, h_t^{dec_1}) \tag{7}$$

Previous analysis showed that the attention over the visual features is inconsistent and weak. We argue that this is because of the diluted relevant visual information, and the competition with the far more relevant source text information.

### 3.3 Filtered Attention (FA)

In order to enhance the visual attention, we propose an extension to the multimodal attention where the objective is to filter the convolutional feature maps using the last hidden state of the source language encoder (Figure 1). We conjecture that a learnable masking operation over the convolutional feature maps can help the decoder-side visual attention mechanism by filtering out regions irrelevant to translation and focus on the most important part of the visual input. The filtered convolutional feature map $\widetilde{\mathrm{V}}$ is computed as follows:

$$\beta^{pre} = ConvAtt(\left[Tile(h_S^{enc}); V\right]) \tag{8}$$

$$\widetilde{\mathrm{V}} = \beta^{pre} \odot \mathrm{V}, \beta^{pre} \in \mathrm{R}^{1\times w \times w} \tag{9}$$

$ConvAtt$ block is inspired from previous works in visual question answering (VQA) (Yang et al., 2016; Kazemi and Elqursh, 2017). It basically computes a spatial attention distribution $\beta^{pre}$ which we further use to mask the actual convolutional features V. The filtered $\widetilde{\mathrm{V}}$ replaces V in the equation 7 instead of being pooled into a single visual embedding in contrast to VQA models.

| EN→DE test2017 | BLEU | METEOR |
|---|---|---|
| Baseline NMT | 31.0 ± 0.3 | 52.1 ± 0.4 |
| $\mathrm{MA}_{448}$ | 28.6 ± 0.8 | 50.1 ± 0.3 |
| $\mathrm{MA}_{448}$ + $\mathrm{L}_2$-norm | 30.8 ± 0.5 | 52.0 ± 0.2 |

Table 1: Impact of $\mathrm{L}_2$ normalization on the performance of multimodal attention.

## 4 Results

We train each model 4 times using different seeds and report mean and standard deviation for the final results using *multeval* (Clark et al., 2011)

**Feature Normalization** We can see from Table 1 that without $\mathrm{L}_2$ normalization, multimodal attention is not able to reach the performance of baseline NMT. Applying the normalization consistently improves the results for all input sizes by around ~2 points in BLEU and METEOR. From now on, we only present systems trained with normalized features.

| EN→DE test2017 | BLEU | METEOR |
|---|---|---|
| $\mathrm{MA}_{224}$ | 30.6 ± 0.4 | 51.8 ± 0.2 |
| $\mathrm{MA}_{448}$ | 30.8 ± 0.5 | 52.0 ± 0.2 |
| $\mathrm{FA}_{224}$ | 31.5 ± 0.5 | 52.2 ± 0.5 |
| $\mathrm{FA}_{448}$ | 31.6 ± 0.5 | 52.5 ± 0.4 |

Table 2: Impact of input image width on the performance of multimodal attention variants.

**Image Size** Although the impact of doubling the image width and height at the input seems marginal (Table 2), we switch to 448x448 images to benefit from the slight gains which are consistent across both attention variants.

### 4.1 Monomodal vs Multimodal Comparison

We first present the mean and standard deviation of BLEU and METEOR over 4 runs on the internal test set test2017 (Table 3). With the help of $\mathrm{L}_2$ normalization, MA system almost reaches the monomodal system but fails to improve over it. On the contrary, the filtered attention (FA) mechanism improves over the baseline and produces hypotheses that are statistically different than the baseline with $p \leq 0.02$.

The improvements obtained for EN→DE language pair are not reflected on the EN→FR performance. One should note that the hyperparameters from EN→DE task were transferred to EN→FR without any other tuning.

The automatic evaluation of our final submissions (which are ensembles of 4 runs) on the official test set test2018 is presented in Table 5. In addition to our submissions, we also provide the best constrained and unconstrained systems[3] in terms of METEOR. However, it should be noted that the submitted systems will be primarily evaluated using human direct assessment.

On EN→DE, our constrained FA system is comparable to the constrained UMONS submission. On EN→FR, our submission obtained the

[3] www.statmt.org/wmt18/multimodal-task.html

| English→German | # Params | `test2017` ($\mu \pm \sigma$) | | |
|---|---|---|---|---|
| | | BLEU | METEOR | TER |
| Baseline NMT | 4.6M | 31.0 ± 0.3 | 52.1 ± 0.4 | 51.2 ± 0.5 |
| Multimodal Attention (MA) | 10.0M | 30.8 ± 0.5 | 52.0 ± 0.2 | 51.1 ± 0.7 |
| Filtered Attention (FA) | 11.3M | **31.6 ± 0.5** | **52.5 ± 0.4** | **50.5 ± 0.5** |

Table 3: EN→DE results: Filtered attention is statistically different than the NMT ($p \leq 0.02$).

| English→French | # Params | `test2017` ($\mu \pm \sigma$) | | |
|---|---|---|---|---|
| | | BLEU | METEOR | TER |
| Baseline NMT | 4.6M | 53.1 ± 0.3 | 69.9 ± 0.2 | 31.9 ± 0.8 |
| Multimodal Attention (MA) | 10.0M | 52.6 ± 0.3 | 69.6 ± 0.3 | 31.9 ± 0.4 |
| Filtered Attention (FA) | 11.3M | 52.8 ± 0.2 | 69.6 ± 0.1 | 31.9 ± 0.1 |

Table 4: EN→FR results: multimodal systems are not able to improve over NMT in terms of automatic metrics.

| EN→DE | BLEU | METEOR | TER |
|---|---|---|---|
| MeMAD† | 38.5 | 56.6 | 44.5 |
| UMONS$\star$ | 31.1 | 51.6 | 53.4 |
| LIUMCVC-FA$\star$ | 31.4 | 51.4 | 52.1 |
| LIUMCVC-NMT$\star$ | 31.1 | 51.5 | 52.6 |
| EN→FR | | | |
| CUNI† | 40.4 | 60.7 | 40.7 |
| LIUMCVC-FA$\star$ | 39.5 | 59.9 | 41.7 |
| LIUMCVC-NMT$\star$ | 39.1 | 59.8 | 41.9 |

Table 5: Official `test2018` results (†: Unconstrained, $\star$: Constrained.)

highest automatic evaluation scores among the constrained submissions and is slightly worse than the unconstrained CUNI system.

## 5 Conclusion

MMT task consists of translating a source sentence into a target language with the help of an image representing the source sentence. The different level of relevance of both input modalities makes it a difficult task where the image should be used with parsimony. With the aim of improving the attention over visual input, we introduced a filtering technique to allow the network to ignore irrelevant parts of the image that should not be considered during decoding. This is done by using an attention-like mechanism between the source sentence and the convolutional feature maps. Results show that this mechanism significantly improves the results for English→German on one of the test sets. In the future, we plan to qualitatively analyze the spatial attention and try to improve it further.

## Acknowledgments

This work was supported by the French National Research Agency (ANR) through the CHIST-ERA M2CR project[4], under the contract number ANR-15-CHR2-0006-01 and by MINECO through APCIN 2015 under the contract number PCIN-2015-251.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost van de Weijer. 2017a. Lium-cvc submissions for wmt17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 432–439, Copenhagen, Denmark. Association for Computational Linguistics.

Ozan Caglayan, Loïc Barrault, and Fethi Bougares. 2016. Multimodal attention for neural machine translation. *CoRR*, abs/1609.03976.

Ozan Caglayan, Mercedes García-Martínez, Adrien Bardet, Walid Aransa, Fethi Bougares, and Loïc Barrault. 2017b. Nmtpy: A flexible toolkit for advanced neural machine translation systems. *Prague Bull. Math. Linguistics*, 109:15–28.

[4] http://m2cr.univ-lemans.fr

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. 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)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. 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*, HLT '11, pages 176–181, Stroudsburg, PA, USA. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the Second Shared Task on Multimodal Machine Translation and Multilingual Image Description. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Desmond Elliott and Ákos Kádár. 2017. Imagination improves multimodal translation. *CoRR*, abs/1705.04350.

K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*.

Jindřich Helcl and Jindřich Libovický. 2017. Cuni system for the wmt17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 450–457, Copenhagen, Denmark. Association for Computational Linguistics.

Vahid Kazemi and Ali Elqursh. 2017. Show, ask, attend, and answer: A strong baseline for visual question answering.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1310–III–1318. JMLR.org.

Ofir Press and Lior Wolf. 2016. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch-Mayne, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Lubli, Antonio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the EACL 2017 Software Demonstrations*, pages 65–68. Association for Computational Linguistics (ACL).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation*, pages 543–553, Berlin, Germany. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2048–2057. JMLR Workshop and Conference Proceedings.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola. 2016. Stacked attention networks for image question answering. In *CVPR*, pages 21–29. IEEE Computer Society.

# The MeMAD Submission to the WMT18 Multimodal Translation Task

**Stig-Arne Grönroos**
Aalto University

**Benoit Huet**
EURECOM

**Mikko Kurimo**
Aalto University

**Jorma Laaksonen**
Aalto University

**Bernard Merialdo**
EURECOM

**Phu Pham**
Aalto University

**Mats Sjöberg**
Aalto University

**Umut Sulubacak**
University of Helsinki

**Jörg Tiedemann**
University of Helsinki

**Raphael Troncy**
EURECOM

**Raúl Vázquez**
University of Helsinki

## Abstract

This paper describes the MeMAD project entry to the WMT Multimodal Machine Translation Shared Task.

We propose adapting the Transformer neural machine translation (NMT) architecture to a multi-modal setting. In this paper, we also describe the preliminary experiments with text-only translation systems leading us up to this choice.

We have the top scoring system for both English-to-German and English-to-French, according to the automatic metrics for *flickr18*.

Our experiments show that the effect of the visual features in our system is small. Our largest gains come from the quality of the underlying text-only NMT system. We find that appropriate use of additional data is effective.

## 1 Introduction

In multi-modal translation, the task is to translate from a source sentence and the image that it describes, into a target sentence in another language. As both automatic image captioning systems and crowd captioning efforts tend to mainly yield descriptions in English, multi-modal translation can be useful for generating descriptions of images for languages other than English. In the MeMAD project[1], multi-modal translation is of interest for creating textual versions or descriptions of audio-visual content. Conversion to text enables both indexing for multi-lingual image and video search, and increased access to the audio-visual materials for visually impaired users.

| Data set | images | en | de | fr | sentences |
|---|---|---|---|---|---|
| Multi30k | ✓ | ✓ | ✓ | ✓ | 29k |
| MS-COCO | ✓ | ✓ | + | + | 616k |
| OpenSubtitles | | ✓ | ✓ | ✓ | 23M/42M |
| | | | | | 1M, 3M, and 6M subsets used. |

Table 1: Summary of data set sizes. ✓means attribute is present in original data. + means data set augmented in this work.

We adapt[2] the Transformer (Vaswani et al., 2017) architecture to use global image features extracted from Detectron, a pre-trained object detection and localization neural network. We use two additional training corpora: MS-COCO (Lin et al., 2014) and OpenSubtitles2018 (Tiedemann, 2009). MS-COCO is multi-modal, but not multi-lingual. We extended it to a synthetic multi-modal and multilingual training set. OpenSubtitles is multilingual, but does not include associated images, and was used as text-only training data. This places our entry in the unconstrained category of the WMT shared task. Details on the architecture used in this work can be found in Section 4.1. Further details on the synthetic data are presented in Section 2. Data sets are summarized in Table 1.

## 2 Experiment 1: Optimizing Text-Based Machine Translation

Our first aim was to select the text-based MT system to base our multi-modal extensions on.

[1] https://www.memad.eu/

[2] Our fork available from https://github.com/Waino/OpenNMT-py/tree/develop_mmod

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 603–611
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64066

| EN-FR | flickr16 | flickr17 | mscoco17 |
|---|---|---|---|
| multi30k | 61.4 | 54.0 | 43.1 |
| +SUBS$_{\text{full}}$ | 53.7 | 48.9 | 47.0 |
| +domain-tuned | 66.1 | 59.7 | **51.7** |
| +ensemble-of-3 | **66.5** | **60.2** | 51.6 |

| EN-DE | flickr16 | flickr17 | mscoco17 |
|---|---|---|---|
| multi30k | 38.9 | 32.0 | 27.7 |
| +SUBS$_{\text{full}}$ | 41.3 | 34.1 | 31.3 |
| +domain-tuned | 43.3 | 38.4 | 35.0 |
| +ensemble-of-3 | **43.9** | **39.6** | **37.0** |

Table 2: Adding subtitle data and domain tuning for image caption translation (BLEU% scores). All results with Marian Amun.

|  | EN-FR | flickr16 | flickr17 | mscoco17 |
|---|---|---|---|---|
| A | SUBS1M$_H$+MS-COCO | 66.3 | 60.5 | 52.1 |
| A | +domain-tuned | 66.8 | 60.6 | 52.0 |
| A | +labels | **67.2** | 60.4 | 51.7 |
| T | SUBS1M$_{LM}$+MS-COCO | 66.9 | 60.3 | **52.8** |
| T | +labels | **67.2** | **60.9** | 52.7 |

|  | EN-DE | flickr16 | flickr17 | mscoco17 |
|---|---|---|---|---|
| A | SUBS1M$_H$+MS-COCO | 43.1 | 39.0 | 35.1 |
| A | +domain-tuned | 43.9 | 39.4 | 35.8 |
| A | +labels | 43.2 | 39.3 | 34.3 |
| T | SUBS1M$_{LM}$+MS-COCO | **44.4** | 39.4 | 35.0 |
| T | +labels | 44.1 | **39.8** | **36.5** |

Table 3: Using automatically translated image captions and domain labels (BLEU% scores). A is short for Amun, T for Transformer.

We tried a wide range of models, but only include results with the two strongest systems: Marian NMT with the *amun* model (Junczys-Dowmunt et al., 2018), and OpenNMT (Klein et al., 2017) with the *Transformer* model.

We also studied the effect of additional training data. Our initial experiments showed that movie subtitles and their translations work rather well to augment the given training data. Therefore, we included parallel subtitles from the OpenSubtitles2018 corpus to train better text-only MT models. For these experiments, we apply the Marian amun model, an attentional encoder-decoder model with bidirectional LSTM's on the encoder side. In our first series of experiments, we observed that domain-tuning is very important when using Marian. The domain-tuning was accomplished by a second training step on in-domain data after training the model on the entire data set. Table 2 shows the scores on development data. We also tried decoding with an ensemble of three independent runs, which also pushed the performance a bit.

Furthermore, we tried to artificially increase the amount of in-domain data by translating existing English image captions to German and French. For this purpose, we used the large MS-COCO data set with its 100,000 images that have five image captions each. We used our best multidomain model (see Table 2) to translate all of those captions and used them as additional training data. This procedure also transfers the knowledge learned by the multidomain model into the caption translations, which helps us to improve the coverage of the system with less out-of-domain data. Hence, we filtered the large collection of translated movie subtitles to a smaller portion of reliable sentence pairs (one million in the experiment we report) and could train on a smaller data set with better results.

We experimented with two filtering methods. Initially, we implemented a basic heuristic filter (SUBS$_H$), and later we improved on this with a language model filter (SUBS$_{LM}$). Both procedures consider each sentence pair, assign it a quality score, and then select the highest scoring 1, 3, or 6 million pairs, discarding the rest. The SUBS$_H$ method counts terminal punctuation (*'.'*, *'...'*, *'?'*, *'!'*) in the source and target sentences, initializing the score as the negative of the absolute value of the difference between these counts. Afterwards, it further decrements the score by 1 for each occurrence of terminal punctuation beyond the first in each of the sentences. The SUBS$_{LM}$ method first preprocesses the data by filtering samples by length and ratio of lengths, applying a rule-based noise filter, removing all characters not present in the Multi30k set, and deduplicating samples. Afterwards, target sentences in the remaining pairs are scored using a character-based deep LSTM language model trained on the Multi30k data. Both selection procedures are intended for noise filtering, and SUBS$_{LM}$ additionally acts as domain adaptation. Table 3 lists the scores we obtained on development data.

To make a distinction between automatically translated captions, subtitle translations and human-translated image captions, we also introduced domain labels that we added as special tokens to the beginning of the input sequence. In this way, the model can use explicit information about the domain when deciding how to translate given input. However, the effect of such labels is not consistent between systems. For Marian amun, the effect is negligible as we can see in Table 3. For the Transformer, domain labels had little effect on BLEU but were clearly beneficial according to chrF-1.0.

### 2.1 Preprocessing of textual data

The final preprocessing pipeline for the textual data consisted of lowercasing, tokenizing using Moses, fixing double-encoded entities and other encoding problems, and normalizing punctuation. For the OpenSubtitles data we additionally used the SUBS$_{LM}$ subset selection.

Subword decoding has become popular in NMT. Careful choice of translation units is especially important as one of the target languages of our system is German, a morphologically rich language. We trained a shared 50k subword vocabulary using Byte Pair Encoding (BPE) (Sennrich et al., 2015). To produce a balanced multi-lingual segmentation, the following procedure was used: First, word counts were calculated individually for English and each of the 3 target languages Czech[3], French and German. The counts were normalized to equalize the sum of the counts for each language. This avoided imbalance in the amount of data skewing the segmentation in favor of some language. Segmentation boundaries around hyphens were forced, overriding the BPE.

Multi-lingual translation with target-language tag was done following Johnson et al. (2016). A special token, e.g. <TO_DE> to mark German as the target language, was prefixed to each paired English source sentence.

## 3 Experiment 2: Adding Automatic Image Captions

Our first attempt to add multi-modal information to the translation model includes the incorporation of automatically created image captions in a purely text-based translation engine. For this, we generated five English captions for each of the images in the provided training and test data. This was done by using our in-house captioning system (Shetty et al., 2018). The image captioning system uses a 2-layer LSTM with residual connections to generate captions based on scene context and object location descriptors, in addition to standard CNN-based features. The model was trained with the MS-COCO training data and used to be state of the art in the COCO leaderboard[4] in Spring 2016. The beam search size was set to five.

| EN-FR | flickr16 | flickr17 | mscoco17 |
|---|---|---|---|
| multi30k | 61.4 | 54.0 | 43.1 |
| +autocap (dual attn.) | 60.9 | 52.9 | 43.3 |
| +autocap 1 (concat) | 61.7 | 53.7 | 43.9 |
| +autocap 1-5 (concat) | **62.2** | **54.4** | **44.1** |
| EN-DE | flickr16 | flickr17 | mscoco17 |
| multi30k | 38.9 | 32.0 | 27.7 |
| +autocap (dual attn.) | 37.8 | 30.2 | 27.0 |
| +autocap 1 (concat) | 39.7 | **32.2** | **28.8** |
| +autocap 1-5 (concat) | **39.9** | 32.0 | 28.7 |

Table 4: Adding automatic image captions (only the best one or all 5). The table shows BLEU scores in %. All results with Marian Amun.

We tried two models for the integration of those captions: (1) a dual attention multi-source model that adds another input sequence with its own decoder attention and (2) a concatenation model that adds auto captions at the end of the original input string separated by a special token. In the second model, attention takes care of learning how to use the additional information and previous work has shown that this, indeed, is possible (Niehues et al., 2016; Östling et al., 2017). For both models, we applied Marian NMT that already includes a working implementation of dual attention translations. Table 4 summarizes the scores on the three development test sets for English-French and English-German.

We can see that the dual attention model does not work at all and the scores slightly drop. The concatenation approach works better probably because the common attention

[3] Czech was later dropped as a target language due to time constraints.

[4] https://competitions.codalab.org/competitions/3221

model learns interactions between the different types of input. However, the improvements are small if any and the model basically learns to ignore the auto captions, which are often very different from the original input. The attention pattern in the example of Figure 1 shows one of the very rare cases where we observe at least some attention to the automatic captions.

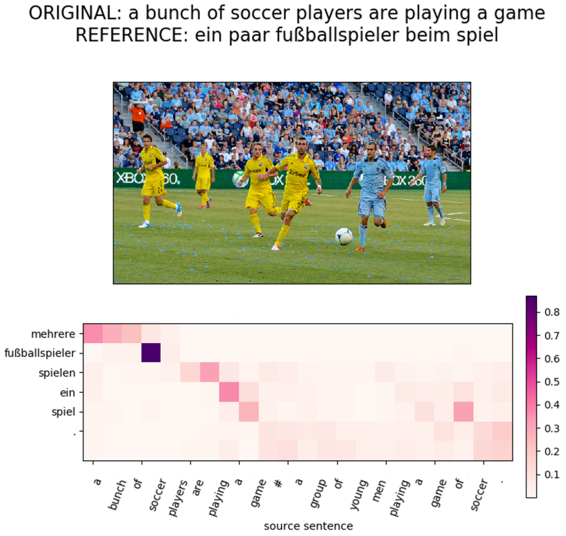


Figure 1: Attention layer visualization for an example where at least one of the attention weights for the last part of the sentence, which corresponds to the automatically generated captions, obtains a value above 0.3

## 4 Experiment 3: Multi-modal Transformer

One benefit of NMT, in addition to its strong performance, is its flexibility in enabling different information sources to be merged. Different strategies to include image features both on the encoder and decoder side have been explored. We are inspired by the recent success of the Transformer architecture to adapt some of these strategies for use with the Transformer.

Recurrent neural networks start their processing from some **initial hidden state**. Normally, a zero vector or a learned parameter vector is used, but the initial hidden state is also a natural location to introduce additional context e.g. from other modalities. Initializing can be applied in either the encoder (IMG$_E$) or decoder (IMG$_D$) (Calixto et al., 2017). These approaches are not directly applicable to the Transformer, as it is not a recurrent model, and lacks a comparable initial hidden state.

**Double attention** is another popular choice, used by e.g. Caglayan et al. (2017). In this approach, two attention mechanisms are used, one for each modality. The attentions can be separate or hierarchical. While it would be possible to use double attention with the Transformer, we did not explore it in this work. The multiple multi-head attention mechanisms in the Transformer leave open many challenges in how this integration would be done.

**Multi-task learning** has also been used, e.g. in the Imagination model (Elliott and Kádár, 2017), where the auxiliary task consists of reconstructing the visual features from the source encoding. Imagination could also have been used with the Transformer, but we did not explore it in this work.

The **source sequence** itself is also a possible location for including the visual information. In the IMG$_W$ approach, the visual features are encoded as a pseudo-word embedding concatenated to the word embeddings of the source sentence. When the encoder is a bidirectional recurrent network, as in Calixto et al. (2017), it is beneficial to add the pseudo-word both at the beginning and the end to make it available for both encoder directions. This is unnecessary in the Transformer, as it has equal access to all parts of the source in the deeper layers of the encoder. Therefore, we add the pseudo-word only to the beginning of the sequence. We use an affine projection of the image features $V \in \mathbb{R}^{80}$ into a pseudo-word embedding $x_I \in \mathbb{R}^{512}$

$$x_I = W_{src} \cdot V + b_I.$$

In the LIUM *trg-mul* (Caglayan et al., 2017), the **target embeddings** and visual features are interacted through elementwise multiplication.

$$y'_j = y_j \odot \tanh(W^{dec}_{mul} \cdot V)$$

Our initial gating approach resembles *trg-mul*.

### 4.1 Architecture

The baseline NMT for this experiment is the OpenNMT implementation of the Transformer. It is an encoder-decoder NMT system using the Transformer architecture (Vaswani et al., 2017) for both the encoder and decoder side. The Transformer is a deep, non-recurrent network for processing variable-length sequences. A Transformer is a stack of layers, consisting of two types of sub-layer: multi-head (MH) attention (Att) sub-layers and feed-forward (FF) sub-layers:

$$\begin{aligned}\mathrm{Att}(Q,K,V) &= \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \\ a_i &= \mathrm{Att}(QW_i^Q, KW_i^K, VW_i^V) \\ \mathrm{MH}(Q,K,V) &= [a_1; \ldots; a_h]W^O \\ \mathrm{FF}(x) &= \max(0, xW_1 + b_1)W_2 + b_2\end{aligned} \tag{1}$$

where $Q$ is the input query, $K$ is the key, and $V$ the attended values. Each sub-layer is individually wrapped in a residual connection and layer normalization.

When used in translation, Transformer layers are stacked into an encoder-decoder structure. In the encoder, the layer consists of a self-attention sub-layer followed by a FF sub-layer. In self-attention, the output of the previous layer is used as queries, keys and values $Q = K = V$. In the decoder, a third context attention sub-layer is inserted between the self-attention and the FF. In context attention, $Q$ is again the output of the previous layer, but $K = V$ is the output of the encoder stack. The decoder self-attention is also masked to prevent access to future information. Sinusoidal position encoding makes word order information available.

**Decoder gate**. Our first approach is inspired by *trg-mul*. A gating layer is introduced to modify the pre-softmax prediction distribution. This allows visual features to directly suppress a part of the output vocabulary. The probability of correctly translating a source word with visually resolvable ambiguity can be increased by suppressing the unwanted choices.

At each timestep the decoder output $s_j$ is projected to an unnormalized distribution over the target vocabulary.

$$y_j = W \cdot s_j + b$$

Before normalizing the distribution using a softmax layer, a gating layer can be added.

| EN-FR | flickr16 | flickr17 | mscoco17 |
|---|---|---|---|
| IMG$_W$ | *68.30* | **62.45** | 52.86 |
| enc-gate | 68.01 | 61.38 | **53.40** |
| dec-gate | 67.99 | 61.53 | 52.38 |
| enc-gate + dec-gate | **68.58** | *62.14* | *52.98* |

| EN-DE | flickr16 | flickr17 | mscoco17 |
|---|---|---|---|
| IMG$_W$ | *45.09* | 40.81 | 36.94 |
| enc-gate | 44.75 | **41.44** | **37.76** |
| dec-gate | **45.21** | 40.79 | 36.47 |
| enc-gate + dec-gate | 44.91 | *41.06* | *37.40* |

Table 5: Comparison of strategies for integrating visual information (BLEU% scores). All results using Transformer, Multi30k+MS-COCO+SUBS3M$_{LM}$, Detectron mask surface, and domain labeling.

$$\begin{aligned} g &= \sigma(W_{gate}^{dec} \cdot V + b_{gate}^{dec}) \\ y'_j &= y_j \odot g \end{aligned} \tag{2}$$

Preliminary experiments showed that gating based on only the visual features did not work. Suppressing the same subword units during the entire decoding of the sentence was too disruptive. We addressed this by using the decoder hidden state as additional input to control the gate. This causes the vocabulary suppression to be time dependent.

$$g_j = \sigma(U_{gate}^{dec} \cdot s_j + W_{gate}^{dec} \cdot V + b_{gate}^{dec}) \tag{3}$$

**Encoder gate**. The same gating procedure can also be applied to the output of the encoder. When using the encoder gate, the encoded source sentence is disambiguated, instead of suppressing part of the output vocabulary.

$$\begin{aligned} g_i &= \sigma(U_{gate}^{enc} \cdot h_i + W_{gate}^{enc} \cdot V + b_{gate}^{enc}) \\ h'_i &= h_i \odot g_i \end{aligned} \tag{4}$$

The gate biases $b_{gate}^{dec}$ and $b_{gate}^{enc}$ should be initialized to positive values, to start training with the gates opened. We also tried combining both forms of gating.

### 4.2 Visual feature selection

Image feature selection was performed using the LIUM-CVC translation system (Caglayan et al., 2017) training on the WMT18 training

| EN-FR | flickr16 | flickr17 | mscoco17 |
|---|---|---|---|
| SUBS3M$_{LM}$ detectron | 68.30 | 62.45 | 52.86 |
| +ensemble-of-3 | 68.72 | 62.70 | 53.06 |
| −visual features | **68.74** | **62.71** | 53.14 |
| −MS-COCO | 67.13 | 61.17 | **53.34** |
| −multi-lingual | 68.21 | 61.99 | 52.40 |
| SUBS6M$_{LM}$ detectron | 68.29 | 61.73 | 53.05 |
| SUBS3M$_{LM}$ gn2048 | 67.74 | 61.78 | 52.76 |
| SUBS3M$_{LM}$ text-only | 67.72 | 61.75 | 53.02 |
| **EN-DE** | **flickr16** | **flickr17** | **mscoco17** |
| SUBS3M$_{LM}$ detectron | 45.09 | 40.81 | 36.94 |
| +ensemble-of-3 | 45.52 | **41.84** | **37.49** |
| −visual features | **45.59** | 41.75 | 37.43 |
| −MS-COCO | 45.11 | 40.52 | 36.47 |
| −multi-lingual | 44.95 | 40.09 | 35.28 |
| SUBS6M$_{LM}$ detectron | 45.50 | 41.01 | 36.81 |
| SUBS3M$_{LM}$ gn2048 | 45.38 | 40.07 | 36.82 |
| SUBS3M$_{LM}$ text-only | 44.87 | 41.27 | 36.59 |
| +multi-modal finetune | 44.56 | 41.61 | 36.93 |

Table 6: Ablation experiments (BLEU% scores). The row SUBS3M$_{LM}$ *detectron* shows our best single model. Individual components or data choices are varied one by one. + stands for adding a component, and − for removing a component or data set. Multiple modifications are indicated by increasing the indentation.

data, and evaluating on the *flickr16, flickr17* and *mscoco17* data sets. This setup is different from our final NMT architecture as the visual feature selection stage was performed at an earlier phase of our experiments. However, the LIUM-CVC setup without training set expansion was also faster to train which enabled a more extensive feature selection process.

We experimented with a set of state-of-the-art visual features, described below.

**CNN-based features** are 2048-dimensional feature vectors produced by applying reverse spatial pyramid pooling on features extracted from the 5$^{th}$ Inception module of the pre-trained GoogLeNet (Szegedy et al., 2015). For a more detailed description, see (Shetty et al., 2018). These features are referred to as gn2048 in Table 6.

**Scene-type features** are 397-dimensional feature vectors representing the association score of an image to each of the scene types in SUN397 (Xiao et al., 2010). Each association score is determined by a separate Radial Basis Function Support Vector Machine (RBF-SVM) classifier trained from pre-trained GoogLeNet CNN features (Shetty et al., 2018).

**Action-type features** are 40-dimensional feature vectors created with RBF-SVM classifiers similarly to the scene-type features, but using the Stanford 40 Actions dataset (Yao et al., 2011) for training the classifiers. Pre-trained GoogLeNet CNN features (Szegedy et al., 2015) were again used as the first-stage visual descriptors.

**Object-type and location features** are generated using the Detectron software[5] which implements Mask R-CNN (He et al., 2017) with ResNeXt-152 (Xie et al., 2017) features. Mask R-CNN is an extension of Faster R-CNN object detection and localization (Ren et al., 2015) that also generates a segmentation mask for each of the detected objects. We generated an 80-dimensional *mask surface* feature vector by expressing the image surface area covered by each of the MS-COCO classes based on the detected masks.

We found that the Detectron mask surface resulted in the best BLEU scores in all evaluation data sets for improving the German translations. Only for *mscoco17* the results could be slightly improved with a fusion of mask surface and the SUN 397 scene-type feature. For French, the results were more varied, but we focused on improving the German translation results as those were poorer overall. We experimented with different ways of introducing the image features into the translation model implemented in LIUM-CVC, and found as in (Caglayan et al., 2017), that *trg-mul* worked best overall.

Later we learned that the *mscoco17* test set has some overlap with the COCO 2017 training set, which was used to train the Detectron models. Thus, the results on that test set may not be entirely reliable. However, we still feel confident in our conclusions as they are also confirmed by the *flickr16* and *flickr17* test sets.

## 4.3 Training

We use the following parameters for the network:[6] 6 Transformer layers in both encoder and decoder, 512-dimensional word embeddings and hidden states, dropout 0.1, batch

[5] https://github.com/facebookresearch/Detectron

[6] Parameters were chosen following the OpenNMT FAQ http://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model

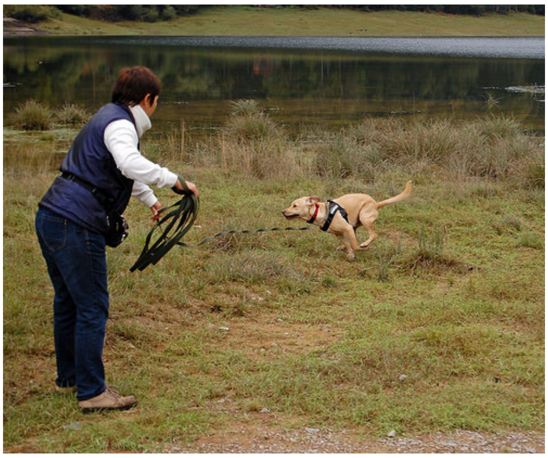


Figure 2: Image 117 was translated correctly as feminine "eine besitzerin steht still und ihr brauner hund rennt auf sie zu ." when not using the image features, but as masculine "ein besitzer ..." when using them. The English text contains the word "her". The person in the image has short hair and is wearing pants.

size 4096 tokens, label smoothing 0.1, Adam with initial learning rate 2 and $\beta_2$ 0.998.

For decoding, we use an ensemble procedure, in which the predictions of 3 independently trained models are combined by averaging after the softmax layer to compute combined prediction.

We evaluate the systems using uncased BLEU using multibleu. During tuning, we also used characterF (Popovic, 2015) with $\beta$ set to 1.0.

There are no images paired with the sentences in OpenSubtitles. When using OpenSubtitles in training multi-modal models, we feed in the mean vector of all visual features in the training data as a dummy visual feature.

### 4.4 Results

Based on the previous experiments, we chose the Transformer architecture, Multi30k+MS-COCO+SUBS3M$_{LM}$ data sets, Detectron mask surface visual features, and domain labeling.

Table 5 shows the BLEU scores for this configuration with different ways of integrating the visual features. The results are inconclusive. The ranking according to chrF-1.0 was not any clearer. Considering the results as a whole and the simplicity of the method, we chose IMG$_W$ going forward.

Table 6 shows results of ablation experiments removing or modifying one component or data choice at a time, and results when using ensemble decoding. Using ensemble decoding gave a consistent but small improvement. Multi-lingual models were clearly better than mono-lingual models. For French, 6M sentences of subtitle data gave worse results than 3M.

We experimented with adding multimodality to a pre-trained text-only system using a fine tuning approach. In the fine tuning phase, a *dec-gate* gating layer was added to the network. The parameters of the main network were frozen, allowing only the added gating layer to be trained. Despite the freezing, the network was still able to unlearn most of the benefits of the additional text-only data. It appears that the output vocabulary was reduced back towards the vocabulary seen in the multi-modal training set. When the experiment was repeated so that the fine-tuning phase included the text-only data, the performance returned to approximately the same level as without tuning (+multi-modal finetune row in Table 6).

To explore the effect of the visual features on the translation of our final model, we performed an experiment where we retranslated using the ensemble while "blinding" the model. Instead of feeding in the actual visual features for the sentence, we used the mean vector of all visual features in the training data. The results are marked *-visual features* in Table 6. The resulting differences in the translated sentences were small, and mostly consisted of minor variations in word order. BLEU scores for French were surprisingly slightly improved by this procedure. We did not find clear examples of successful disambiguation. Figure 2 shows one example of a detrimental use of visual features.

It is possible that adding to the training data forward translations of MS-COCO captions from a text-only translation system introduced a biasing effect. If there is translational ambiguity that should be resolved using the image, the text-only system will not be able to resolve it correctly, instead likely yielding the word that is most frequent in that textual context. Using such data for training a multi-modal system might bias it towards ignoring the image.

On this year's *flickr18* test set, our system scores 38.54 BLEU for English-to-German and 44.11 BLEU for English-to-French.

## 5 Conclusions

Although we saw an improvement from incorporating multi-modal information, the improvement is modest. The largest differences in quality between the systems we experimented with can be attributed to the quality of the underlying text-only NMT system.

We found the amount of in-domain training data and multi-modal training data to be of great importance. The synthetic MS-COCO data was still beneficial, despite being forward translated, and the visual features being overconfident due to being extracted from a part of the image classifier training data.

Even after expansion with synthetic data, the available multi-modal data is dwarfed by the amount of text-only data. We found that movie subtitles worked well for this purpose. When adding text-only data, domain adaptation was important, and increasing the size of the selection met with diminishing returns.

Current methods do not fully address the problem of how to efficiently learn from both large text-only data and small multi-modal data simultaneously. We experimented with a fine tuning approach to this problem, without success.

Although the effect of the multi-modal information was modest, our system still had the highest performance of the task participants for the English-to-German and English-to-French language pairs, with absolute differences of +6.0 and +3.5 BLEU%, respectively.

## Acknowledgments

This work has been supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 780069, and by the Academy of Finland in the project 313988. In addition the Finnish IT Center for Science (CSC) provided computational resources. We would also like to acknowledge the support by NVIDIA and their GPU grant.

## References

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost Van de Weijer. 2017. LIUM-CVC submissions for WMT17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation*.

Iacer Calixto, Koel Dutta Chowdhury, and Qun Liu. 2017. DCU system report on the WMT 2017 multi-modal machine translation task. In *Proceedings of the Second Conference on Machine Translation*. pages 440–444.

Desmond Elliott and Àkos Kádár. 2017. Imagination improves multimodal translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. volume 1, pages 130–141.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, pages 2980–2988.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558* .

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*. Association for Computational Linguistics, Melbourne, Australia, pages 116–121. http://www.aclweb.org/anthology/P18-4020.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*. https://doi.org/10.18653/v1/P17-4012.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR* abs/1405.0312. http://arxiv.org/abs/1405.0312.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. pages 1828–1836.

Robert Östling, Yves Scherrer, Jörg Tiedemann, Gongbo Tang, and Tommi Nieminen. 2017. The helsinki neural machine translation system. In *Proceedings of the Second Conference on Machine Translation*. pages 338–347.

Maja Popovic. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *WMT15*. pages 392–395.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems (NIPS)*. pages 91–99.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. In *ACL16*.

Rakshith Shetty, Hamed Rezazadegan Tavakoli, and Jorma Laaksonen. 2018. Image and video captioning with augmented neural architectures. *IEEE MultiMedia* To appear.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 1–9.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria, volume V, pages 237–248.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. pages 6000–6010.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE conference on Computer vision and pattern recognition (CVPR)*. IEEE, pages 3485–3492.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pages 5987–5995.

Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas J. Guibas, and Fei-Fei Li. 2011. Human action recognition by learning bases of action attributes and parts. In *International Conference on Computer Vision (ICCV)*. Barcelona, Spain, pages 1331–1338.

# The AFRL-Ohio State WMT18 Multimodal System: Combining Visual with Traditional

**Jeremy Gwinnup**
AFRL
jeremy.gwinnup.1
@us.af.mil

**Joshua Sandvick**
Ohio State University
sandvick.6
@osu.edu

**Michael Hutt**
AFRL
michael.hutt.ctr
@us.af.mil

**Grant Erdmann**
AFRL
grant.erdmann
@us.af.mil

**John U. Duselis**
AFRL
john.duselis@us.af.mil

**James W. Davis**
Ohio State University
davis.1719@osu.edu

## Abstract

AFRL-Ohio State extends its usage of visual domain-driven machine translation for use as a peer with traditional machine translation systems. As a peer, it is enveloped into a system combination of neural and statistical MT systems to present a composite translation.

## 1 Introduction

Most of the submissions to the Second Conference on Machine Translation (WMT17) Multimodal submissions for Task 1 (Elliott et al., 2017) used the visual domain to enhance machine translation of the image+caption pair. The exception was a Visual Machine Translation (VMT) system where the image is the driver for the translation (Duselis et al., 2017). While the scores for this submission did not approach baseline, except by human scoring, it did introduce the concept that the visual domain can approach parity with the traditional text based MT systems.

The AFRL-Ohio State Third Conference on Machine Translation (WMT18) submission also explores viability of a VMT system enhancing current techniques. Previous work by Calixto et al. (2017) ensembled different multimodal machine translation (MMT) systems, with the visual domain used in conjunction with the text domain. Similarly, we incorporate the VMT system with a small sampling of neural and statistical MT systems in order to give indicators on how the performance is affected by mutual inclusion.

## 2 The AFRL-Ohio State 2018 Multimodal System Submission

A Visual Machine Translation system is one that utilizes the visual domain, whether it is a video or picture, as the driver for MT. This assumes that there is a visual analogue for the relevant source text. This is a specialized form of Multimodal Machine Translation (MT) in which the image is producing candidate target language sentences.

Current trends in MT use system combinations or ensembles of various MT systems (statistical, neural, rule-based, etc.) to create a consensual final answer. A key ingredient to this method is introducing variability of MT outputs to reach the conclusion (Freitag et al., 2014). We posit that adding the VMT to the system will enhance the overall results.

AFRL-Ohio State submitted three systems for official scoring. The focus of explanation will be on the 4Combo system because it underwent human evaluation, but the other two will be revisited in the analysis portion. No post-editing was performed for any of the submission systems.

### 2.1 The AFRL-Ohio State WMT17 Submission

Here is an overview of the VMT system submitted to the WMT17 submission (Duselis et al., 2017). This system architecture assumes a captionator can be trained in a target language to give meaningful output in the form of a set of the most probable $n$ target language candidate captions. A learned mapping function of the encoded source language caption to the corresponding encoded target language candidate captions is thus employed. Finally, a distance function is applied, and the nearest candidate caption is selected to be the translation of the source caption.

### 2.2 Captionator

The current instantiation of our VMT system uses the Google Show and Tell captionator (Vinyals et al., 2015) trained on the training set from Flickr30k, augmented with data from ImageCLEF 2008 (Grubinger et al., 2006).

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 612–615
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64067

The captionator was trained on the 29,000 training image+German caption pairs, plus 20,000 image+German captions from ImageCLEF 2008. This was slightly fewer than the number used on the WMT17 submission. Additional models were trained on the constrained set of the 29,000 WMT pairs, one with a single caption per image and another with five captions per image. However, the Show and Tell system generated a high number of 'unknown word' tokens. Filtering out the sentences with unknown tokens produced a bias towards short, generic captions. Augmenting with the ImageCLEF data produced noticeably better results. This was the only change for the captionator. Consistent with the prior year's submission, no accommodations were made for out of vocabulary words.

### 2.3 Caption Selection

Stemming from critique and results from WMT17, the simple neural network was revised to center around a two sided Long Short Term Memory (LSTM) encoder. One side of the LSTM was trained to encode English sentences, while the other was trained to encode German sentences. Each of the LSTMs has a state size of 256 nodes. The multiclass hinge loss function was used to evaluate the encodings, penalizing the loss by the highest-scoring incorrect match between the English and German sentences in a training batch.

The training data comprised the WMT18 Multimodal Task 1 English and German training sentences from the 2018 Multi30k dataset. The words were tokenized and transformed to lower case, and punctuation was removed. Words were then embedded using the FastText pretrained word embedding vectors (Bojanowski et al., 2017), with dimension 300. The Adam optimizer (Kingma and Ba, 2014) was employed to train the network parameters with a batch size of 32. The network was trained for approximately 100 epochs using TensorFlow on a GeForce GTX 1080.

We tested the caption selection mechanism on the 2018 Multi30k datasets, encoding both the given English captions and the given German captions. Each English caption was matched with the German caption in the set with minimum hinge loss. On the 29,000-image training set, each English caption was correctly matched with its corresponding German caption 99.4% of the time. On the 1,014-image test set, the matching accuracy was 92.4%.

### 2.4 "Standard" Machine Translation

Inspired by Gwinnup et al. (2017), we trained multiple MT systems with differing toolkits and characteristics for use in system combination with our VMT efforts. These toolkits include: OpenNMT (Klein et al., 2017), Marian (Junczys-Dowmunt et al., 2018), and Moses (Koehn et al., 2007).

All systems were trained with the approximately 41 million parallel lines of preprocessed German–English data provided by the WMT18 organizers.

#### 2.4.1 OpenNMT

The OpenNMT system was trained using the German-English Parallel Data from the WMT18 organizers for the News Task, but excluding the ParaCrawl Data. It incorporates case features and a vocabulary from 2000 byte-pair encoding merges. This small vocabulary was chosen to reduce the number of out-of-vocabulary tokens resulting from morphology and compounding.

#### 2.4.2 Marian

The Marian toolkit was used to train a baseline system using the pre-BPE'd data provided by the WMT18 news task organizers. This system employed a deep bi-directional (or "BiDeep") architecture as outlined in Miceli Barone et al. (2017) and Sennrich et al. (2017). Further details of the exact settings used to train this system are available in the wmt2017-uedin example shown in the marian-examples GitHub repository[1].

#### 2.4.3 Moses

For variety, a phrase-based Moses system was trained using the same BPE'd data as the above Marian system. This system employed a hierarchical reordering model (Galley and Manning, 2008), 5-gram operation sequence model (Durrani et al., 2011) and a 5-gram BPE'd KenLM (Heafield, 2011) language model trained on the target side of the provided parallel data.

### 2.5 System Combination

RWTH's Jane System combination (Freitag et al., 2014) was used to combine the outputs of the three traditional MT systems with the output of our VMT approach.

[1] https://github.com/marian-nmt/marian-examples/tree/master/wmt2017-uedin

## 3 Analysis

### 3.1 Results

The AFRL-Ohio State WMT18 4Combo submission, although a better showing than our WMT17 submission, failed to meet baseline. Comparing the VMT component to last year's system showed the expected improvement in results. The official results are presented in Table 3.1, mirroring the results presented in Specia et al. (2018). VMT is the visually driven MT system. 2Combo is the VMT+Marian, 3Combo is the Marian+Moses+OpenNMT system. 4Combo is all four systems.

| System | BLEU ↑ | Meteor ↑ | TER ↓ |
|---|---|---|---|
| VMT | 5.0 | 17.7 | 80.1 |
| 2Combo | 10.0 | 25.4 | 79.0 |
| 3Combo | 23.8 | 44.5 | 59.7 |
| 4Combo | 24.3 | 45.4 | 58.6 |

Table 1: Systems Scoring

Examining the 3Combo and 4Combo outputs, we note a positive performance trend when adding the VMT system to combinations of traditional MT systems.

#### 3.1.1 Captionator Output - Oracle Scoring

To gain more insight, a document level Meteor and BLEU Oracle scoring for the captionator output was applied.

The three observables were the most probable sentence from the captionator, the AFRL-Ohio State caption selection mechanism, and the best scoring caption output. This analysis is based on the WMT17 multimodal validation set.

We performed *a posteriori* analysis, to determine how well our caption selector compares with other possibilities. We considered two options. First, the one-best is the caption the captionator considers the most likely, without regard to the source-side text. Second, we found an oracle caption for each image, based on Meteor score. The oracle captions determine an upper-bound on the Meteor score the caption selector can achieve. Results are shown in Table 2.

## 4 Future Work

Our purpose in developing the visual domain is to include it as an equal to the text or as a driver for the MT at a higher level of abstraction than the neural layer. Using the captionator to produce sentences limits the VMT to the the captionator's abilities. Instead, we next plan to employ a more generalized approach to estimate objects or concepts that are particularly difficult to translate directly from the image (or video clip, if available) rather than attempting to estimate an actual sentence structure. We expect the use of such information from the visual content to be more amenable to bias or influence other MT systems.

| Method | BLEU ↑ | METEOR ↑ |
|---|---|---|
| 1-best | 1.53 | 10.69 |
| LSTM | 5.74 | 18.59 |
| Oracle | 18.78 | 36.74 |

Table 2: Oracle scoring for the VMT system.

## References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017. Doubly-Attentive Decoder for Multi-modal Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1924, Vancouver, Canada. Association for Computational Linguistics.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics (ACL '11)*, pages 1045–1054, Portland, Oregon.

John Duselis, Michael Hutt, Jeremy Gwinnup, James Davis, and Joshua Sandvick. 2017. The afrl-osu wmt17 multimodal translation system: An image processing approach. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 445–449, Copenhagen, Denmark. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 215–233, Copenhagen, Denmark. Association for Computational Linguistics.

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 9 Aug 2018. Originator reference number RH-18-118708. Case number 88ABW-2018-3967.

Markus Freitag, Matthias Huck, and Hermann Ney. 2014. Jane: Open source machine translation system combination. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32, Gothenburg, Sweden.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 848–856.

Michael Grubinger, Paul D. Clough, Henning Müller, , and Thomas Deselaers. 2006. The iapr bencmark: A new evaluation resource for visual information systems. In *International Conference on Language Resources and Evaluation*.

Jeremy Gwinnup, Timothy Anderson, Grant Erdmann, Katherine Young, Michaeel Kazi, Elizabeth Salesky, Brian Thompson, and Jonathan Taylor. 2017. The AFRL-MITLL WMT17 systems: Old, New, Borrowed, BLEU. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 303–309, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, Melbourne, Australia.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180.

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep architectures for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 99–107. Association for Computational Linguistics.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Lucia Specia, Stella Frank, Loïc Barrault, Fethi Bougares, and Desmond Elliot. 2018. WMT18 shared task: Multimodal machine translation.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

# CUNI System for the WMT18 Multimodal Translation Task

**Jindřich Helcl** and **Jindřich Libovický** and **Dušan Variš**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
{helcl, libovicky, varis}@ufal.mff.cuni.cz

## Abstract

We present our submission to the WMT18 Multimodal Translation Task. The main feature of our submission is applying a self-attentive network instead of a recurrent neural network. We evaluate two methods of incorporating the visual features in the model: first, we include the image representation as another input to the network; second, we train the model to predict the visual features and use it as an auxiliary objective. For our submission, we acquired both textual and multimodal additional data. Both of the proposed methods yield significant improvements over recurrent networks and self-attentive textual baselines.

## 1 Introduction

Multimodal Machine Translation (MMT) is one of the tasks that seek ways of capturing the relation of texts in different languages given a shared "grounding" information in a different (e.g. visual) modality.

The goal of the MMT shared task is to generate an image description (caption) in the target language using a caption in the source language and the image itself. The main motivation for this task is the development of models that can exploit the visual information for meaning disambiguation and thus model the denotation of words.

During the last years, MMT was addressed as a subtask of neural machine translation (NMT). It was thoroughly studied within the framework of recurrent neural networks (RNNs) (Specia et al., 2016; Elliott et al., 2017). Recently, the architectures based on self-attention such as the Transformer (Vaswani et al., 2017) became state-of-the-art in NMT.

In this work, we present our submission based on the Transformer model. We propose two ways of extending the model. First, we tweak the architecture such that it is able to process both modalities in a multi-source learning scenario. Second, we leave the model architecture intact, but add another training objective and train the textual encoder to be able to predict the visual features of the image described by the text. This training component has been introduced in RNNs by Elliott and Kádár (2017) and is called the "imagination".

We find that with self-attentive networks, we are able to improve over a strong textual baseline by including the visual information in the model. This has been proven challenging in the previous RNN-based submissions, where there was only a minor difference in performance between textual and multimodal models (Helcl and Libovický, 2017; Caglayan et al., 2017).

This paper is organized as follows. Section 2 summarizes the previous submissions and related work. In Section 3, we describe the proposed methods. The details of the datasets used for the training are given in Section 4. Section 5 describes the conducted experiments. We discuss the results in Section 6 and conclude in Section 7.

## 2 Related Work

Currently, most of the work has been done within the framework of sequence-to-sequence learning. Although some of the proposed approaches use explicit image analysis (Shah et al., 2016; Huang et al., 2016), most methods use image representation obtained using image classification networks pre-trained on ImageNet (Deng et al., 2009), usually VGG19 (Simonyan and Zisserman, 2014) or ResNet (He et al., 2016a).

In the simplest case, the image can be represented as a single vector from the penultimate layer of the image classification network. This vector can be then plugged in at various places of the sequence-to-sequence architecture (Libovický et al., 2016; Calixto and Liu, 2017).

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 616–623
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64068

Several methods compute visual context information as a weighted sum over the image spatial representation using the attention mechanism (Bahdanau et al., 2014; Xu et al., 2015) and combine it with the context vector from the textual encoder in doubly-attentive decoders. Caglayan et al. (2016) use the visual context vector in a gating mechanism applied to the textual context vector. Caglayan et al. (2017) concatenate the context vectors from both modalities. Libovický and Helcl (2017) proposed advanced strategies for computing a joint attention distribution over the text and image. We follow this approach in our first proposed method described in Section 3.1.

The visual information can also be used as an auxiliary objective in a multi-task learning setup. Elliott and Kádár (2017) propose an imagination component that predicts the visual features of an image from the textual encoder representation, effectively regularizing the encoder part of the network. The imagination component is trained using a maximum margin objective. We reuse this approach in our method described in Section 3.2.

## 3 Architecture

We examine two methods of exploiting the visual information in the Transformer architecture. First, we add another encoder-decoder attention layer to the decoder which operates over the image features directly. Second, we train the network with an auxiliary objective using the imagination component as proposed by Elliott and Kádár (2017).

### 3.1 Doubly Attentive Transformer

The Transformer network follows the encoder-decoder scheme. Both parts consist of a number of layers. Each encoder layer first attends to the previous layer using self-attention, and then applies a single-hidden-layer feed-forward network to the outputs. All layers are interconnected with residual connections and their outputs are normalized by layer normalization (Ba et al., 2016). A decoder layer differs from an encoder layer in two aspects. First, as the decoder operates autoregressively, the self-attention has to be masked to prevent the decoder to attend to the "future" states. Second, there is an additional attention sub-layer applied after self-attention which attends to the final states of the encoder (called *encoder-decoder*, or *cross* attention).

The key feature of the Transformer model is the use of attention mechanism instead of recurrence relation in RNNs. The attention can be conceptualized as a soft-lookup function that operates on an associative array. For a given set of queries $Q$, the attention uses a similarity function to compare each query with a set of keys $K$. The resulting similarities are normalized and used as weights to compute a context vector which is a weighted sum over a set of values $V$ associated with the keys. In self-attention, all the queries, keys and values correspond to the set of states of the previous layer. In the following cross-attention sub-layer, the set of resulting context vectors from the self-attention sub-layer is used as queries, and keys and values are the states of the final layer of the encoder.

The Transformer uses scaled dot-product as a similarity metric for both self-attention and cross-attention. For a query matrix $Q$, key matrix $K$ and value matrix $V$, and the model dimension $d$, we have:

$$\mathcal{A}(Q, K, V) = \operatorname{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V. \quad (1)$$

The attention is used in a multi-head setup. This means that we first linearly project the queries, keys, and values into a number of smaller matrices and then apply the attention function $\mathcal{A}$ independently on these projections. The set of resulting context vectors $C$ is computed as a sum of the outputs of each attention head, linearly projected to the original dimension:

$$C = \sum_{i=1}^{h} \mathcal{A}(QW_i^Q, KW_i^K, VW_i^V)W_i^O \quad (2)$$

where $(W_i^O)^\top$, $W_i^Q$, $W_i^K$, and $W_i^V \in \mathbb{R}^{d \times d_h}$ are trainable parameters, $d$ is the dimension of the model, $h$ is the number of heads, and $d_h$ is a dimension of a single head. Note that despite $K$ and $V$ being identical matrices, the projections are trained independently.

In this method, we introduce the visual information to the model as another encoder via an additional cross-attention sub-layer. The keys and values of this cross-attention correspond to the vectors in the last convolutional layer of a pre-trained image processing network applied on the input image. This sub-layer is inserted between the textual cross-attention and the feed-forward network, as illustrated in Figure 1. The set of the context vectors from the textual cross-attention is used as

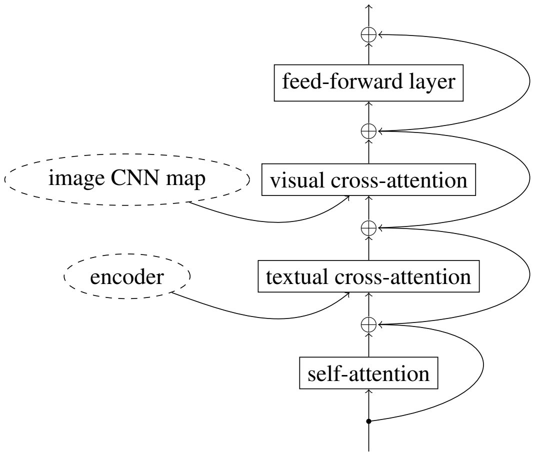


Figure 1: One layer of the doubly-attentive Transformer decoder with 4 sub-layers connected with residual connections.

queries, and the context vectors of the visual cross-attention are used as inputs to the feed-forward sub-layer. Similarly to the other sub-layers, the input is linked to the output by a residual connection. Equation 3 shows the computation of the visual context vectors given trainable matrices $Z_i^Q$, $Z_i^K$, $Z_i^V$, and $Z_i^O$ for $i = 1, \ldots, h$; the set of textual context vectors is denoted by $C_{txt}$ and the extracted set of image features as $F$:

$$C_{img} = \sum_{i=1}^{h} \mathcal{A}(C_{txt}Z_i^Q, FZ_i^K, FZ_i^V)Z_i^O. \quad (3)$$

### 3.2 Imagination

We use the imagination component of Elliott and Kádár (2017) originally proposed for training multimodal translation models using RNNs. We adapt it in a straightforward way in our Transformer-based models.

The imagination component serves effectively as a regularizer to the encoder, making it consider the visual meaning together with the words in the source sentence. This is achieved by training the model to predict the image representations that correspond to those computed by a pre-trained image classification network. Given a set of encoder states $h_j$, the model computes the predicted image representation as follows:

$$\hat{y}_{img} = W_2^R \max(0, W_1^R \textstyle\sum_j h_j) \quad (4)$$

where $W_1^R \in \mathbb{R}^{r\times d}$ and $W_2^R \in \mathbb{R}^{n\times r}$ are trainable parameter matrices, $d$ is the Transformer model dimension, $r$ is a hidden layer dimension of the imagination component, and $n$ is the dimension of the image feature vector. Note that Equation 4 corresponds to a single-hidden-layer feed-forward network with a ReLU activation function applied on the sum of the encoder states.

We train the visual feature predictor using an auxiliary objective. Since the encoder part of the model is shared, additional weight updates are propagated to the encoder during the model optimization w.r.t. this additional loss. For the generated image representation $\hat{y}$ and the reference representation $y$, the error is estimated as margin-based loss with margin parameter $\alpha$:

$$\mathcal{L}_{imag} = \max\left(0, \alpha + d(\hat{y}, y) - d(\hat{y}, y_c)\right) \quad (5)$$

where $y_c$ is a contrastive example randomly drawn from the training batch and $d$ is a distance function between the representation vectors, in our case the cosine distance.

Unlike Elliott and Kádár (2017), we sum both translation and imagination losses within the training batches rather than alternating between training of each component separately.

| | en | de | fr | cs |
|---|---|---|---|---|
| Training | 29,000 sentences | | | |
| Tokens | 378k | 361k | 410k | 297k |
| Average length | 13.0 | 12.4 | 14.1 | 10.2 |
| # tokens range | 4–40 | 2–44 | 4–55 | 2–39 |
| Validation | 1,014 sentences | | | |
| Tokens | 13k | 13k | 14k | 10k |
| Average length | 13.1 | 12.7 | 14.2 | 10.2 |
| # tokens range | 4–30 | 3–33 | 5–36 | 4–27 |
| OOV rate | 1.28% | 3.09% | 1.20% | 3.95% |

Table 1: Multi30k statistics on training and validation data – total number of tokens, average number of tokens per sentence, and lengths of the shortest and the longest sentence.

## 4 Data

The participants were provided with the Multi30k dataset (Elliott et al., 2016), an extension of the Flickr30k dataset (Plummer et al., 2017) which contains 29,000 train images, 1,014 validation images and 1,000 test images. The images are accompanied with six captions which were independently obtained through crowd-sourcing. In

Multi30k, each image is accompanied also with German, French, and Czech translations of a single English caption. Table 1 shows statistics of the captions contained in the Multi30k dataset.

Since the Multi30k dataset is relatively small, we acquired additional data, similarly to our last year submission (Helcl and Libovický, 2017). The overview of the dataset structure is given in Table 2.

First, for German only, we prepared synthetic data out of the WMT16 MMT Task 2 training dataset using back-translation to English (Sennrich et al., 2016). This data consists of five additional German descriptions of each image. Along with the data for Task 1 which is the same as the training data this year, the back-translated part of the dataset contains 174k sentences.

Second, for Czech and German, we selected pseudo in-domain data by filtering the available general domain corpora. For both languages, we trained a character-level RNN language model on the corresponding language parts of the Multi30k training data. We use a single layer bidirectional LSTM (Hochreiter and Schmidhuber, 1997) network with 512 hidden units and character embeddings with dimension of 128. For Czech, we compute perplexities of the Czech sentences in the CzEng corpus (Bojar et al., 2016b). We selected 15k low-perplexity sentence pairs out of 64M sentence pairs in total by setting the perplexity threshold to 2.5. For German, we used the additional data from the last year (Helcl and Libovický, 2017), which was selected out of several parallel corpora (EU Bookshop (Skadiņš et al., 2014), News Commentary (Tiedemann, 2012) and CommonCrawl (Smith et al., 2013)).

Third, also for Czech and German, we applied the same criterion on monolingual corpora and used back-translation to create synthetic parallel data. For Czech, we took 333M sentences of CommonCrawl and 66M sentences of News Crawl (which is used in the WMT News Translation Task; Bojar et al., 2016a) and extracted 18k and 11k sentences from these datasets respectively.

Finally, we use the whole EU Bookshop as an additional out-of-domain parallel data. Since the size of this dataset is large relative to the sizes of the other parts, we oversample the rest of the data to balance the in-domain and out-of-domain portions of the training dataset. The oversampling factors are shown in Table 2.

| | de | fr | cs |
|---|---|---|---|
| Multi30k | 29k | | |
| – oversampling factor | 273× | 366× | 9× |
| Task 2 BT | 145k | — | — |
| in-domain parallel | 3k | — | 15k |
| in-domain BT | 30k | — | 29k |
| – oversampling factor | 39× | — | 7× |
| EU Bookshop | 9.3M | 10.6M | 445k |
| COCO (English only) | 414k | | |

Table 2: Overview of the data used for training our models with oversampling factors. The EU Bookshop data was not oversampled. BT stands for back-translation.

For the unconstrained training of the imagination component, we used the MSCOCO (Lin et al., 2014) dataset which consists of 414k images along with English captions.

## 5 Experiments

In this year's round, two variants of the MMT tasks were announced. As in the previous years, the goal of Task 1 is to translate an English caption into the target language given the image. The target languages are German, French and Czech. In Task 1a, the model receives the image and its captions in English, German, and French and is trained to produce the Czech translation. In our submission, we focus only on Task 1.

In our submission, we experiment with three distinct architectures. First, in *textual* architectures, we leave out the images from the training altogether. We use this as a strong baseline for the multimodal experiments. Second, *multimodal* experiments use the doubly attentive Transformer decoder described in Section 3.1. Third, the experiments referred to as *imagination* employ the imagination component as described in Section 3.2.

We train the models in constrained and unconstrained setups. In the constrained setup, only the Multi30k dataset is used for training. In the unconstrained setup, we train the model using the additional data described in Section 4. We run the multimodal experiments only in the constrained setup.

In the unconstrained variant of the imagination experiments, the dataset consists of examples that can miss either the textual target values (MSCOCO extension), or the image (additional

| | | en-cs | | en-fr | | en-de | |
|---|---|---|---|---|---|---|---|
| | | single | averaged | single | averaged | single | averaged |
| Caglayan et al. (2017) | | N/A | | 54.7/71.3 | 56.7/73.0 | 37.8/57.7 | 41.0/**60.5** |
| Cons. | Textual | 29.6/28.9 | 30.9/29.5 | 59.2/73.7 | 59.7/74.4 | 38.1/56.2 | 38.3/56.0 |
| | Imagniation | 29.8/29.4 | 30.5/29.6 | 59.4/74.2 | 59.7/74.4 | 38.8/56.4 | 39.2/56.8 |
| | Multimodal | 30.5/29.7 | 31.0/29.9 | 60.6/75.0 | 60.8/75.1 | 38.4/53.1 | 38.7/57.2 |
| Unc. | Textual | 31.2/30.1 | 32.3/30.7 | 62.0/76.7 | 62.5/76.7 | 39.6/58.7 | 40.4/59.0 |
| | Imagination | **36.3/32.8** | 35.9/32.7 | **62.8/77.0** | **62.8/77.0** | **42.7**/59.1 | 42.6/59.4 |

Table 3: Results on the 2016 test set in terms of BLEU score and METEOR score. We compare our results with the last year's best system (Caglayan et al., 2017) which used model ensembling instead of weight averaging.

parallel data). In these cases, we train only the decoding component with specified target value (i.e. imagination component on visual features, or the Transformer decoder on the textual data). As said in Section 3.2, we train both components by summing the losses when both the image and the target sentence are available in a training example.

In all experiments, we use the Transformer network with 6 layers with model dimension of 512 and feed-forward hidden layer dimension of 4096 units. The embedding matrix is shared between the encoder and decoder and its transposition is reused as the output projection matrix (Press and Wolf, 2017). For each language pair, we use a vocabulary of approximately 15k wordpieces (Wu et al., 2016). We extract the vocabulary and train the model on lower-cased text without any further pre-processing steps applied. We tokenize the text using the algorithm bundled with the tensor2tensor library (Vaswani et al., 2018). The tokenization algorithm splits the sentence to groups of alphanumeric and non-alphanumeric groups, throwing away single spaces that occur inside the sentence. We conduct the experiments using the Neural Monkey toolkit (Helcl and Libovický, 2017).[1]

For image pre-processing, we use ResNet-50 (He et al., 2016a) with identity mappings (He et al., 2016b). In the doubly-attentive model, we use the outputs of the last convolutional layer before applying the activation function with dimensionality of $8 \times 8 \times 2048$. We apply a trainable linear projection to the maps into 512 dimensions to fit the Transformer model dimension. In the imagination experiments, we use average-pooled maps with 2048 dimensions. Following Elliott and Kádár (2017), we set the margin parameter $\alpha$ from Equation 5 to 0.1.

For each model, we keep 10 sets of parameters that achieve the best BLEU scores (Papineni et al., 2002) on the validation set. We experiment with weight averaging and model ensembling. However, these methods performed similarly and we thus report only the results of the weight averaging, which is computationally less demanding.

In all experiments, we use the Adam optimizer (Kingma and Ba, 2014) with initial learning rate 0.2, and Noam learning rate decay scheme (Vaswani et al., 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$ and 4,000 warm-up steps.

## 6 Results

We report the quantitative results of measured on the Multi30k 2016 test set in Table 3.

The Transformer architecture achieves generally comparable or better results than the RNN-based architecture. Adding the visual information has a significant positive effect on the system performance, both when explicitly provided as a model input and when used as an auxiliary objective. In the constrained setup which used only the data from the Multi30k dataset, the doubly-attentive decoder performed best.

The biggest gain in performance was achieved by training on the additional parallel data. The imagination architecture outperforms the purely textual models.

As the performance of single models increases, the positive effect of weight averaging diminishes. The effect of checkpoint averaging is smaller than the results reported by Caglayan et al. (2017) who use ensembles of multiple models trained with a different initialization – we use only checkpoints from a single training run.

During the qualitative analysis, we noticed that mostly for Czech target language, the systems are

[1] `https://github.com/ufal/neuralmonkey`

often incapable of capturing morphology. In order to quantify this, we also measured the BLEU scores using the lemmatized system outputs and references. The difference was around 4 BLEU points for Czech, less than 3 BLEU points for French, and around 2 BLEU points for German. These differences were consistent among different types of models.

We hypothesize that in the imagination experiments, the visual information is used to learn a better representation of the textual input, which eventually leads to improvements in the translation quality. In the multimodal experiments, the improvements can come from the refining of the textual representation rather than from explicitly using the image as an input.

In order to determine whether the visual information is used also at the inference time, we performed an adversarial evaluation by providing the trained multimodal model with randomly selected "fake" images. In French and Czech, BLEU scores dropped by more than 1 BLEU point. This suggests that the multimodal models utilize the visual information at the inference time as well. The German models seem to be virtually unaffected. We hypothesize this might be due to a different methodology of acquiring the training data for German and the other two target languages (Elliott et al., 2016).

## 7 Conclusions

In our submission for the WMT18 Multimodal Translation Task, we experimented with the Transformer architecture for MMT. The experiments show that the Transformer architecture outperforms the RNN-based models.

Experiments with a doubly-attentive decoder showed that explicit incorporation of visual information improves the model performance. The adversarial evaluation confirms that the models also take into account the visual information.

The best translation quality was achieved by extending the training data by additional image captioning data and parallel textual data. It this unconstrained setup, the best scoring model employs the imagination component that was previously introduced in RNN-based sequence-to-sequence models.

## Acknowledgements

This research received support from the Czech Science Foundation grant no. P103/12/G084, and the grants No. 976518 and 1140218 of the Grant Agency of the Charles University. This research was partially supported by SVV project number 260 453.

## References

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016a. Findings of the 2016 conference on machine translation (WMT16). In *Proceedings of the First Conference on Machine Translation (WMT). Volume 2: Shared Task Papers*, volume 2, pages 131–198, Stroudsburg, PA, USA. Association for Computational Linguistics, Association for Computational Linguistics.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016b. CzEng 1.6: Enlarged Czech-English parallel corpus with processing tools dockered. In *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Computer Science, pages 231–238, Cham / Heidelberg / New York / Dordrecht / London. Masaryk University, Springer International Publishing.

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost van de Weijer. 2017. Lium-cvc submissions for wmt17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 432–439, Copenhagen, Denmark. Association for Computational Linguistics.

Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost van de Weijer. 2016. Does multimodality help human and machine for translation and image captioning? In *Proceedings of the First Conference on Machine Translation*, pages 627–633, Berlin, Germany. Association for Computational Linguistics.

Iacer Calixto and Qun Liu. 2017. Incorporating global visual features into attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 992–1003. Association for Computational Linguistics.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255, Miami, FL, USA. IEEE.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. *CoRR*, abs/1605.00459.

Desmond Elliott and Ákos Kádár. 2017. Imagination improves multimodal translation. *CoRR*, abs/1705.04350.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016a. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016b. Identity mappings in deep residual networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 630–645.

Jindřich Helcl and Jindřich Libovický. 2017. CUNI system for the WMT17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation*, pages 450–457. Association for Computational Linguistics.

Jindřich Helcl and Jindřich Libovický. 2017. Neural Monkey: An open-source tool for sequence learning. *The Prague Bulletin of Mathematical Linguistics*, 107:5–17.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9:1735–1780.

Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. 2016. Attention-based multimodal neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 639–645, Berlin, Germany. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vancouver, Canada. Association for Computational Linguistics.

Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Ondřej Bojar, and Pavel Pecina. 2016. CUNI system for WMT16 automatic post-editing and multimodal translation tasks. In *Proceedings of the First Conference on Machine Translation*, pages 646–654, Berlin, Germany. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2017. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *Int. J. Comput. Vision*, 123(1):74–93.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Kashif Shah, Josiah Wang, and Lucia Specia. 2016. Shef-multimodal: Grounding machine translation on images. In *Proceedings of the First Conference on Machine Translation*, pages 660–665, Berlin, Germany. Association for Computational Linguistics.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne. 2014. Billions of parallel words for free: Building and using the eu bookshop corpus. In

*Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jason R. Smith, Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1383, Sofia, Bulgaria. Association for Computational Linguistics.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation*, pages 543–553, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2048–2057, Lille, France. JMLR Workshop and Conference Proceedings.

# Sheffield Submissions for WMT18 Multimodal Translation Shared Task

**Chiraag Lala, Pranava Madhyastha, Carolina Scarton** and **Lucia Specia**
Department of Computer Science, University of Sheffield, UK
{clala1, p.madhyastha, c.scarton, l.specia}@sheffield.ac.uk

## Abstract

This paper describes the University of Sheffield's submissions to the WMT18 Multimodal Machine Translation shared task. We participated in both tasks 1 and 1b. For task 1, we build on a standard sequence to sequence attention-based neural machine translation system (NMT) and investigate the utility of multimodal re-ranking approaches. More specifically, $n$-best translation candidates from this system are re-ranked using novel multimodal cross-lingual word sense disambiguation models. For task 1b, we explore three approaches: (i) re-ranking based on cross-lingual word sense disambiguation (as for task 1), (ii) re-ranking based on consensus of NMT $n$-best lists from German-Czech, French-Czech and English-Czech systems, and (iii) data augmentation by generating English source data through machine translation from French to English and from German to English followed by hypothesis selection using a multimodal-reranker.

## 1 Introduction

This paper describes the University of Sheffield's submissions for both Tasks 1 and 1b of the third edition of the Multimodal Machine Translation shared task. Task 1 consists in translating source sentences in English that describe an image into German (DE) or French (FR) or Czech (CS), given the image. Task 1b consists in translating source sentences in English that describe an image into Czech, given the image and the French and German translations of the source sentence.

This task poses the challenging problem of building models that use both language and image modalities. The dataset for the shared task (Specia et al., 2016) has sentences with simple language constructions and it has been observed by earlier systems (Specia et al., 2016; Elliott et al., 2017) that standard text-only sequence to sequence neural machine translation models (NMT) with attention are able to obtain very high performance.

Building on this, for further inspection, we built our own standard NMT systems for EN-DE, EN-FR and EN-CS language directions and noticed that the translation hypotheses besides the 1-best output are also of high quality. We made our systems produce 20 translation hypotheses for English descriptions in the validation set and selected the hypothesis with the highest sentence-level METEOR (Denkowski and Lavie, 2014) score, called the Oracle, and compared this to the 1-best. In this experiment, we observed that the Oracle performs way better (11 to 13.5 METEOR points) than the 1-best output (See Table 1). This preliminary experiment motivated us to investigate re-ranking approaches.

| Lang-Pair | 1-best | Best of 20best (Oracle) | Scope/difference (Oracle - 1-best) |
|---|---|---|---|
| EN-DE | 48.36 | 61.85 | **+13.49** |
| EN-FR | 64.91 | 76.87 | **+11.96** |
| EN-CS | 33.87 | 44.71 | **+10.84** |

Table 1: Motivation for re-ranking. In this preliminary experiment, we observe that re-ranking of the 20-best translation hypotheses generated by a standard NMT model has the potential of improving translation by upto 10.84 to 13.49 METEOR points for the three language pairs.

For a re-ranking strategy, we were inspired by how humans use images to translate image descriptions. We believe humans look at the image usually to disambiguate ambiguous words in the source sentence especially in those instances where the text alone is not sufficient. For example, translating '*A* ***sportsperson*** *is playing football*' into French requires us to know whether the sportsperson is a male or a female and accordingly

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 624–631
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64069

the translation is '*Une **sportif** joue au football*' (male) or '*Une **sportive** joue au football*' (female). In such cases, humans usually look at the image to disambiguate and select the correct translation which is what we try to mimic in our approach.

More specifically, in our systems we adopt a two-step pipeline approach. In the first step, we use an ensemble of text-only models initialized with different seeds to produce lists of 10-best translation hypotheses. In the second step, we re-rank the 10-best hypotheses using a novel multimodal cross-lingual Word Sense Disambiguation (WSD) approach. For control experiments, we also compare our results with monomodal cross-lingual WSD (Lefever and Hoste, 2013) and a system that performs re-ranking using the Most Frequent Sense (MFS) baseline (Section 3.1.2).

Our main goal is to investigate a multimodal, image-based, cross-lingual WSD that predicts the translation candidate which correctly disambiguates ambiguous words in the source sentence.

Our baseline NMT system is based on the attentive encoder-decoder (Bahdanau et al., 2015) approach with a Conditional GRU (CGRU) (Cho et al., 2014) decoder and is built using NMTPY toolkit (Caglayan et al., 2017b).

Our cross-lingual WSD models are based on neural sequence learning models for WSD (Raganato et al., 2017; Yuan et al., 2016; Kågebäck and Salomonsson, 2016) applied to the Multimodal Lexical Translation Dataset (Lala and Specia, 2018).

For task 1b, we explore three approaches. The first approach concatenates the 10-best translation hypotheses from DE-CS, FR-CS and EN-CS MT systems and then re-ranks them using the *image-aware* multimodal cross-lingual WSD mentioned earlier (the same way as in Task 1) (Section 3.1.2).

The second approach explores the consensus between the different 10-best lists. The best hypothesis is selected according to the number of times it appeared in the different 10-best lists. We followed the order of the $n$-best lists, meaning that the highest ranked hypothesis with the majority votes was selected.

The third approach uses data augmentation that hinges on the fact that the objective is to translate from English into Czech. Extra source data is generated by building systems that translate from German into English and French into English. With this extra data, we build an EN-CS system. We then obtain a 10-best list over training, development and test sets respectively. For selecting the best hypothesis from the 10-best list, we experiment with a classification-based approach. We calculate METEOR (Denkowski and Lavie, 2014) scores for each hypothesis in the 10-best list of the training set and threshold the scores to build classifiers to distinguish good from bad translations using a) word embeddings and image features with a Random Forest model and b) a multimodal Recurrent Neural Network (RNN) model.

In Section 3 we describe our systems in detail. We describe the data preprocessing in Section 2. The results are discussed in Section 4.

## 2 Data

### 2.1 Translation models

We use the Multi30K (Elliott et al., 2016) dataset provided by the organizers. Each image $i$ contains one English description $en_i$ taken from Flickr30K and human translations into German $de_i$, French $fr_i$ and Czech $cz_i$. In other words, each instance is a 5-tuple of the form $(i, en_i, de_i, fr_i, cz_i)$. The dataset contains 29,000 training and 1,014 development instances.

For Task 1, the test sets of the previous two editions (2016 and 2017) have also been provided for validation purposes. These do not contain Czech translations. A new test set of 1,071 tuples containing an English description and its corresponding image is provided for evaluation.

For Task 1b, a test set of 1,000 tuples containing English, French, and German descriptions and their corresponding images is provided for evaluation. This test set corresponds to the unseen portion of the Czech Test 2017 data. The test set of 2016 is provided for validation purposes.

### 2.2 Cross-lingual WSD models

For the cross-lingual WSD models, we use the Multimodal Lexical Translation Dataset (MLTD) (Lala and Specia, 2018), which was extracted from the Multi30K (Elliott et al., 2016) dataset. MLTD consists of 4-tuples of the form $(x, i, en_i, x_t)$ where $x$ is an ambiguous[1] word in the English description $en_i$ of the image $i$, and $x_t$ is the lexical translation of $x$ in a specified target language $t \in$

[1] We use the term 'ambiguous' for those words in the source language that have multiple translations in the target language in the training portion of the given parallel corpus, where these translations represent different 'senses' of the word in that corpus.

{German, French, Czech} that conforms with the image and the description. Only instances from the training portion of the Multi30K dataset are used to train the cross-lingual WSD models.

For English-German, MLTD consists of 745 ambiguous words in English with 4.09 different translations per word (on average) in German and 17.69 instances per translation (on average) totalling 53,868 MLTD instances.

For English-French, MLTD consists of 661 ambiguous words in English with 2.98 different translations per word (on average) in French and 22.73 instances per translation (on average) totalling 44,779 MLTD instances.

For English-Czech[2], MLTD consists of 3,217 ambiguous words in English with 5.15 different translations per word (on average) in Czech and 11.32 instances per translation (on average) totalling 187,495 MLTD instances.

### 2.3 Image features

We used the ResNet-50 image features provided by the task organizers. These are 2048-dimensional features extracted from *pool5* of a pretrained ResNet-50 (He et al., 2016) model which has been trained on the ImageNet dataset (Russakovsky et al., 2015).

## 3 System descriptions

In this section we describe the systems submitted for both tasks.

### 3.1 Task 1 systems

Our two-step pipeline consists in first obtaining high quality hypotheses from a NMT model, followed by a re-ranking step. We describe the setup of the NMT in Section 3.1.1. The cross-lingual WSD models used for re-ranking are described in Section 3.1.2 and the re-ranking formulation with examples is shown in Section 3.1.3.

#### 3.1.1 Baseline NMT model setup

We make use of an ensemble of text only attention based NMT models (Bahdanau et al., 2015) with a conditional gated recurrent units (CGRU) (Cho et al., 2014) decoder. We build the system using the NMTPY toolkit (Caglayan et al., 2017b). Our models have a setting similar to Caglayan et al. (2016) with a bi-directional 256-dimensional recurrent GRU followed by a conditional GRU which is initialized with a non-linear transformation of the mean of encoder states. We use a simple feedforward network to compute the attention scores as described in Caglayan et al. (2016). We use Adam optimizer with a learning rate of 5e$^{-}$5 and a batch size of 64. We set the embedding dimensionality of encoder and decoder to 128 and follow the default parametrization in (Caglayan et al., 2017a). Our final baseline model is an ensemble of different runs of the model with five different seeds.

#### 3.1.2 Crosslingual WSD models

The goal of cross-lingual WSD (Lefever and Hoste, 2013) is to generate contextually correct translations of ambiguous words in the source language into the target language. For this, the sense inventory for the ambiguous words is created from the parallel corpus. MLTD (Lala and Specia, 2018) (Section 2.2) provides us with the data settings needed for this task.

As a baseline we have the **Most Frequent Sense** (MFS) model, which returns the most frequent translation of a given ambiguous word as seen in the training corpus. For example in the English-French MLTD, the ambiguous word *woods* appears 95 times in the training set. In 16 times the translation is *forêt* (forest), while in the remaining 79 times the translation is *bois* (timber/wood). In this case, the MFS model translates the word *woods* as *bois* irrespective of the context.

As a second baseline, we have a text-only **Lexical Translation** (LT) model. This is a single layer Bidirectional Long Short-Term Memory (BiLSTM) network (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005) used as a sequence tagger as depicted in Figure 1.

For the LT model, we convert the classification task of cross-lingual WSD into a sequence tagging task as demonstrated in (Raganato et al., 2017). The 4-tuples of MLTD are transformed into a sequential tagged dataset. This consists of English sentences where each word is tagged to itself if it is unambiguous and tagged to the correct lexical translation in the target language if it is ambiguous.[3]

[2] This dataset has been extracted using the same procedure in Lala and Specia (2018) except the human filtering step and thus it contains noise: mainly, the multiple "senses" can sometimes correspond to morphological variants or synonym words.

[3] We tried a few more data settings - like each word tagged to 'NA' if it is unambiguous - but these did not result in any improvements.

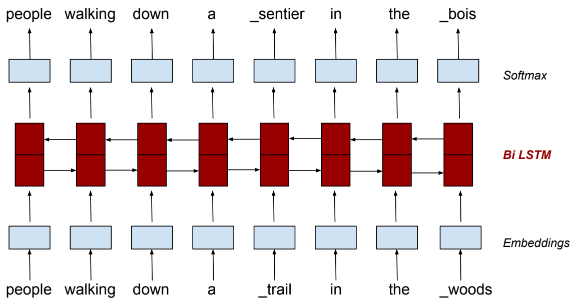


Figure 1: Lexical Translation (LT) model - A BiLSTM that tags each input word in the source sentence. The training is done such that an unambiguous word is tagged with itself, while an ambiguous word, like *trail* and *woods* in this example, is tagged with the corresponding lexical translation in the target language like *sentier* and *bois* respectively.

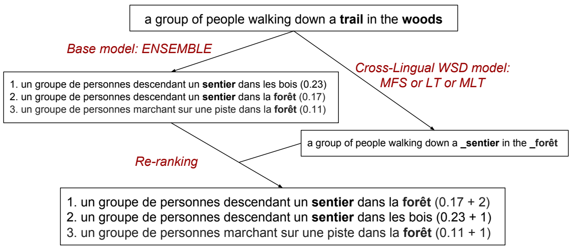
Our proposed model is a **Multimodal Lexical Translation** (MLT) model. It has the same architecture as the LT model except that the LSTM weights are initialized with the image features. [4] To avoid dimensionality mismatch, the image features (Section 2.3) undergo a dimensionality reduction via a fully connected layer, which is also trained.

**Training:** Both LT and MLT models are trained on only those sentences which have at least one ambiguous word as per MLTD. For optimization, we use the ADAM (Kingma and Ba, 2014) algorithm with a learning rate = 0.001 and batch size = 32. The LSTM hidden state dimensions and the word embedding dimensions are set to 300 and the dropout rate is set to 0.3. Training is stopped early if model accuracy over the validation set does not improve for 30 epochs. These models are implemented and trained in the TensorFlow framework.

The performance of the models (Table 2)[5], measured in terms of percentage of correctly translated ambiguous words (accuracy), suggests that the image-aware MLT model is slightly better than the text-only LT and MFS models.

| | test17flickr | test17coco | test16 | train | val |
|---|---|---|---|---|---|
| **EN-DE** | | | | | |
| MFS | 60.47 | 52.49 | 65.34 | 68.93 | 70.25 |
| LT | **61.40** | 57.22 | 69.61 | 79.71 | 67.77 |
| MLT | 59.68 | **57.48** | **69.79** | **80.18** | **68.85** |
| **EN-FR** | | | | | |
| MFS | **77.29** | 67.12 | 77.73 | 78.38 | 79.33 |
| LT | 76.83 | 70.52 | 80.35 | 88.05 | **81.15** |
| MLT | 75.20 | **70.75** | **80.43** | **88.44** | 80.87 |

Table 2: Performance of cross-lingual WSD models (Section 3.1.2) measured in terms of accuracy: proportion of correctly translated ambiguous words.


a group of people walking down a **trail** in the **woods**

1. un groupe de personnes descendant un **sentier** dans les bois (0.23)
2. un groupe de personnes descendant un **sentier** dans la **forêt** (0.17)
3. un groupe de personnes marchant sur une piste dans la **forêt** (0.11)

a group of people walking down a **_sentier** in the **_forêt**

1. un groupe de personnes descendant un **sentier** dans la **forêt** (0.17 + 2)
2. un groupe de personnes descendant un **sentier** dans les bois (0.23 + 1)
3. un groupe de personnes marchant sur une piste dans la **forêt** (0.11 + 1)

Figure 2: Task 1 system pipeline. The base model generates $n$-best translation candidates of the source sentence. The cross-lingual WSD model translates ambiguous words in the source sentence. The re-ranking step uses these lexical translations to re-score the translation candidates.

### 3.1.3 Re-ranking

Our re-ranking strategy is depicted in Figure 2. First, given an English source sentence, the base model generates an $n$-best list of translation candidates with a likelihood score. The idea is to select the translation candidate in the $n$-best translations which correctly disambiguates as many ambiguous words in the source sentence as possible.

The source sentence in our example (Figure 2) contains two ambiguous words *trail* and *woods* as per the English-French MLTD. We use a cross-lingual WSD model, MFS or LT or MLT, to predict the lexical translations of these words (the correct ones being *sentier* and *forêt* respectively in this example). Next, we match these to the words in the translation candidates and add the number of matching words to the original score[6] of the candidates. Then, the $n$-best translations are re-ranked using the new scores and the top candidate (which has the highest number of matches) is used in the evaluation.

[4] We tried a few other ways of using the image features - like concatenating it to word embeddings, using it as a separate word, etc. - but these did not result in any improvements.

[5] The performance of cross-lingual WSD models for EN-CS language direction could not be evaluated because the EN-CS Multimodal Lexical Translation Dataset was noisy. The clean 'filtered by human' versions of the EN-CS MLTD test sets were not ready at the time of submitting this paper.

[6] The likelihood score assigned to the candidate by the baseline NMT model

### 3.2 Task 1b systems

Three different approaches were explored in our submissions for Task 1b. The first approach follows the re-raking experiments using MLT for Task 1. The second approach exploits consensus-based selection and the third explores data augmentation and $n$-best selection through classification. We try two different types of classifiers - Random Forest and Recurrent Neural Network.

**Re-ranking using MLT** For the re-ranking approach, we first train three baseline EN-CS, DE-CS and FR-CS NMT models. Given a source sentence in the test set, we generate 10-best translation hypotheses using each of the three models. The three 10-best lists are concatenated to form a list of 30 translation hypotheses. We then use the trained EN-CS MLT model for cross-lingual WSD and perform re-ranking as mentioned in 3.1.2 and 3.1.3.

**Consensus-based selection** For the consensus-based selection approach, we again use the three 10-best translation hypotheses coming from the EN-CS, DE-CS and FR-CS systems. We then explore consensus between the different 10-best lists. The best hypothesis is selected according to the number of times it appears in the different lists. We follow the order of the EN-CS 10-best list: the highest ranked hypothesis in the EN-CS list with the majority of the votes (measured in terms of whether it occurs in the DE-CS and FR-CS 10-best lists) is selected.

**Data augmentation** We explore data augmentation by creating systems that first translate source sentences from French, German and Czech into English. This leads to variants of the source data that translate into the same Czech sentence. The augmented data is used to train an NMT system to translate test source sentences from English into Czech. We then obtain a 10-best list for the training, development and test sets. For the selection approach, we compute METEOR scores for each of the hypotheses in the 10-best list of the training set. To treat this as a binary classification task, we set a threshold such that the top four hypotheses are assumed to be the best translations and are chosen as positive samples, with the remaining six as bad examples.[7] This is then used to train two types of classifiers:

- Random Forest (RF) classifier: we use the image vectors concatenated with sentence embeddings from source and target sentences as features for training the classifier. For extracting sentence embeddings, we use the approach of Arora et al. (2016). Pre-trained embeddings for English and Czech from MUSE[8] (Conneau et al., 2018) are used.The RF algorithm in the scikit-learn framework (Pedregosa et al., 2011) is trained to distinguish between good and bad translations.
- RNN classifier: We use a simple RNN-based classifier where the last hidden state of the encoded sentence is concatenated with the image vector and used with a hinge loss to distinguish between good and bad translations.

## 4 Results

For both tasks, the initial evaluation was performed in terms of METEOR, BLEU (Papineni et al., 2002) and TER (Snover et al., 2006), with METEOR as the primary metric. Direct human assessments of translation adequacy will be used for the final evaluation by the task organizers.

For task 1, our submitted systems consisted of: a) SHEF_LT: re-ranking using LT model; b) SHEF_MLT: re-ranking using MLT model; c) SHEF_MFS: re-ranking using MFS model; and d) SHEF_Baseline: our baseline text-only ensemble NMT model

Table 3 shows the official evaluation results of our systems submitted to Task 1 and the baseline system provided by the organizers. For all language pairs, our systems outperform the official baseline for all metrics.

For EN-DE and EN-FR, the systems with LT and MLT are slightly better than the system with MFS. For EN-CS, however, the MFS system scores better than the LT and MLT variants. This is, perhaps, because the EN-CS MLTD (on which LT and MLT models are trained) is noisy, as previously mentioned. The dataset has been extracted using the same procedure in (Lala and Specia, 2018) except for the human filtering step, which is crucial for a clean dataset.

On further inspection, we observe that the cross-lingual WSD re-ranking affects only 127 to

[7] This threshold was empirically defined.

[8] https://github.com/facebookresearch/MUSE

| | EN-DE | | | EN-FR | | | EN-CS | | |
|---|---|---|---|---|---|---|---|---|---|
| | METEOR | BLEU | TER | METEOR | BLEU | TER | METEOR | BLEU | TER |
| SHEF_LT | 50.7 | 30.5 | 53.0 | 59.8 | 38.8 | 41.5 | 29.1 | 28.3 | 51.7 |
| SHEF_MLT | 50.7 | 30.4 | 52.9 | 59.8 | 38.9 | 41.5 | 29.1 | 28.2 | 51.7 |
| SHEF_Baseline | 50.7 | 30.9 | 52.4 | 59.8 | 38.9 | 41.2 | 29.4 | 29.0 | 51.1 |
| SHEF_MFS | 50.7 | 30.3 | 53.1 | 59.7 | 38.8 | 41.6 | 29.2 | 27.8 | 52.4 |
| Baseline | 47.4 | 27.6 | 55.2 | 56.9 | 36.3 | 41.6 | 27.7 | 26.5 | 54.4 |

Table 3: Evaluation of our systems and the baseline for Task 1. We show METEOR, BLEU and TER scores.

| | MFS | LT | MLT |
|---|---|---|---|
| EN-DE | 189 (239) | 149 (200) | 148 (189) |
| EN-FR | 163 (244) | 127 (180) | 129 (192) |
| EN-CS | 484 (649) | 100 (124) | 124 (148) |

Table 4: The effect of re-ranking approaches on the baseline NMT model outputs. The number outside the bracket shows the number of instances that are affected due to re-ranking in the 1071 test instances. The number inside the bracket '()' shows the number of words in the entire test set that are affected (deleted, added or replaced) due to re-ranking.

| | METEOR | BLEU | TER |
|---|---|---|---|
| SHEF_CON | 27.6 | 24.7 | 52.1 |
| SHEF_MLT | 27.5 | 24.5 | 52.5 |
| SHEF_ARNN | 27.5 | 25.2 | 53.9 |
| SHEF_ARF | 27.1 | 24.1 | 54.6 |
| Baseline | 26.8 | 23.6 | 54.2 |

Table 5: Evaluation of our systems and the baseline for Task 1b.

189 test instances (for EN-DE and EN-FR only[9]) out of the total 1,071 test instances (See Table 4). These usually result in changing only one or two words and as a result it affects only 180 to 244 words in the entire test set (See Table 4). In other words, only 1.4% words in the entire test set are affected by the re-ranking, which may explain why the performance of all the systems is so similar. It also suggests that automatics metrics like BLEU, METEOR and TER may not be sufficient to detect subtle changes in translation quality making it difficult to deduce insights from our re-ranking approaches. We hope to rely on Direct Human Assessment and other more sensitive metrics to help to better understand the affects.

For Task 1b, we submitted four models: a) SHEF_CON: consensus based model; b) SHEF_MLT: a re-ranking approach using MLT model; c) SHEF_ARNN: a data augmentation and hypothesis selection approach using an RNN classifier; and d) SHEF_ARF: data augmentation and hypothesis selection approach using an RF classifier.

Table 5 shows the automatic metric scores for our systems and the official baseline. Our systems outperform the baseline in terms of BLEU and METEOR. For TER, all systems are better than the baseline except for SHEF_ARF. Our best performing system is SHEF_CON.

## 5 Conclusions

We have described our submissions to the Multimodal Machine Translation shared task at WMT18. We explored novel multimodal $n$-best re-ranking approaches for task 1, and consensus-based approaches for task 1b using image information for re-ranking of an augmented $n$-best list with outputs from different translation models.

All our models perform better than the official baseline for all metrics and language pairs in task 1. However, we observe that SHEF_LT and SHEF_MLT, for the dataset and in the current setup, are not significantly different and their performance are nearly identical which indicates that the image information is not contributing significantly for this task and cross-lingual WSD is, perhaps, not very useful. On the other hand, it is worth emphasising that the corpora used may not show many ambiguous words and our model is not expected to be beneficial in this case.

For task 1b, our models also outperform the official baseline, with the best model being SHEF_CON. As for task 1, the use of image information do not lead to improvements when evaluated using automatic metrics METEOR, BLEU and TER.

[9] We ignore EN-CS in this observation because the EN-CS MLTD is noisy and thus the trained cross-lingual WSD models are not reliable for this language pair.

## Acknowledgements

This work is supported by the MultiMT project (H2020 ERC Starting Grant No. 678017).

## References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings. *In proceedings of International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *In proceedings of International Conference on Learning Representations*.

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost Van de Weijer. 2017a. Lium-cvc submissions for wmt17 multimodal translation task. *In proceedings of Conference on Machine Translation (WMT)*.

Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost van de Weijer. 2016. Does multimodality help human and machine for translation and image captioning? *In proceedings of the First Conference on Machine Translation*.

Ozan Caglayan, Mercedes García-Martínez, Adrien Bardet, Walid Aransa, Fethi Bougares, and Loïc Barrault. 2017b. Nmtpy: A flexible toolkit for advanced neural machine translation systems. *In proceedings of The Prague Bulletin of Mathematical Linguistics*.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. *In proceedings of the International Conference on Learning Representations*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. *In proceedings of the Ninth Workshop on Statistical Machine Translation*.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. *In proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. *In proceedings of Workshop on Vision and Language*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *In proceedings of Neural Networks*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *In proceedings of the IEEE conference on computer vision and pattern recognition*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *In proceedings of Neural Computation*.

Mikael Kågebäck and Hans Salomonsson. 2016. Word sense disambiguation using a bidirectional lstm. *In proceedings of International Conference on Computational Linguistics*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *In proceedings of International Conference on Learning Representations*.

Chiraag Lala and Lucia Specia. 2018. Multimodal Lexical Translation. *In proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Els Lefever and Véronique Hoste. 2013. Semeval-2013 task 10: Cross-lingual word sense disambiguation. *In proceedings of the Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. *In proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

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. 2011. Scikit-learn: Machine learning in Python. *In proceedings of the Journal of Machine Learning Research*.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017. Neural sequence learning models for word sense disambiguation. *In proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *In proceedings of International Journal of Computer Vision*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. *In proceedings of the Seventh Biennial Conference of the Association for Machine Translation in the Americas*.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. *In proceedings of the First Conference on Machine Translation*.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. *In proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*.

# Ensemble Sequence Level Training for Multimodal MT: OSU-Baidu WMT18 Multimodal Machine Translation System Report

**Renjie Zheng** *1 **Yilin Yang** *1 **Mingbo Ma** †1,2 **Liang Huang** †2,1
[1]School of EECS, Oregon State University, Corvallis, OR
[2]Baidu Research, Sunnyvale, CA
zheng@renj.me {yilinyang721, cosmmb, liang.huang.sh}@gmail.com

## Abstract

This paper describes multimodal machine translation systems developed jointly by Oregon State University and Baidu Research for WMT 2018 Shared Task on multimodal translation. In this paper, we introduce a simple approach to incorporate image information by feeding image features to the decoder side. We also explore different sequence level training methods including scheduled sampling and reinforcement learning which lead to substantial improvements. Our systems ensemble several models using different architectures and training methods and achieve the best performance for three subtasks: En-De and En-Cs in task 1 and (En+De+Fr)-Cs task 1B.

## 1 Introduction

In recent years, neural text generation has attracted much attention due to its impressive generation accuracy and wide applicability. In addition to demonstrating compelling results for machine translation (Sutskever et al., 2014; Bahdanau et al., 2014), by simple adaptation, similar models have also proven to be successful for summarization (Rush et al., 2015; Nallapati et al., 2016), image or video captioning (Venugopalan et al., 2015; Xu et al., 2015) and multimodal machine translation (Elliott et al., 2017; Caglayan et al., 2017; Calixto and Liu, 2017; Ma et al., 2017), which aims to translate the caption from one language to another with the help of the corresponding image.

However, the conventional neural text generation models suffer from two major drawbacks. First, they are typically trained by predicting the next word given the previous ground-truth word. But at test time, the models recurrently feed their own predictions into it. This "exposure bias" (Ranzato et al., 2015) leads to error accumulation during generation at test time. Second, the models are optimized by maximizing the probability of the next ground-truth words which is different from the desired non-differentiable evaluation metrics, e.g. BLEU.

Several approaches have been proposed to tackle the previous problems. Bengio et al. (2015) propose scheduled sampling to alleviate "exposure bias" by feeding back the model's own predictions with a slowly increasing probability during training. Furthermore, reinforcement learning (Sutton et al., 1998) is proven to be helpful to directly optimize the evaluation metrics in neural text generation models training. Ranzato et al. (2015) successfully use the REINFORCE algorithm to directly optimize the evaluation metric over multiple text generation tasks. Rennie et al. (2017); Liu et al. (2017) achieve state-of-the-art on image captioning using REINFORCE with baseline to reduce training variance.

Moreover, many existing works show that neural text generation models can benefit from model ensembling by simply averaging the outputs of different models (Elliott et al., 2017; Rennie et al., 2017). Garmash and Monz (2016) claim that it is essential to introduce diverse models into the ensemble. To this end, we ensemble models with various architectures and training methods.

This paper describes our participation in the WMT 2018 multimodal tasks. Our submitted systems include a series of models which only consider text information, as well as multimodal models which also include image information to initialize the decoders. We train these models using scheduled sampling and reinforcement learning. The final outputs are decoded by ensembling those models. To the best of our knowledge, this is the first multimodal machine translation system that achieves the state-of-the-art using sequence level learning methods.

[*] Equal contribution
[†] Contributions made while at Baidu Research

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 632–636
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64070

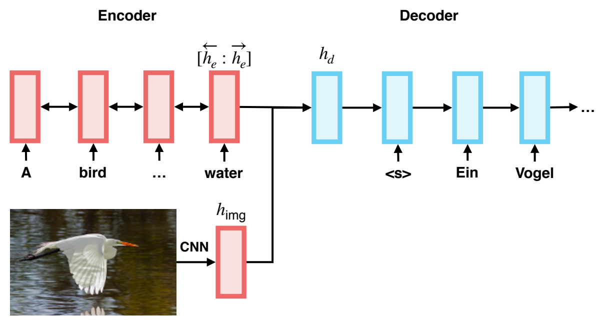


Figure 1: Multimodal Machine Translation Model

## 2 Methods

Our model is based on the sequence-to-sequence RNN architecture with attention (Bahdanau et al., 2014). We incorporate image features to initialize the decoder's hidden state as shown in Figure 1. Originally, this hidden state is initialized using the concatenation of last encoder's forward and backward hidden states, $\overrightarrow{h_e}$ and $\overleftarrow{h_e}$ resp. We propose to use the sum of encoder's output and image features $h_{\text{img}}$ to initialize the decoder. Formally, we have the final initialization state $h_d$ as:

$$h_d = \tanh(W_e[\overrightarrow{h_e}; \overleftarrow{h_e}] + W_{\text{img}} h_{\text{img}} + b). \quad (1)$$

where $W_e$ and $W_{\text{img}}$ project the encoder and image feature vector into the decoder hidden state dimensionality and $b$ is the bias parameter. This approach has been previously explored by Calixto and Liu (2017).

As discussed previously, translation systems are traditionally trained using cross entropy loss. To overcome the discrepancy between training and inference distributions, we train our models using scheduled sampling (Bengio et al., 2015) which mixes the ground truth with model predictions, further adopting the REINFORCE algorithm with baseline to directly optimize translation metrics.

### 2.1 Scheduled Sampling

When predicting a token $\hat{y}_t$, scheduled sampling uses the previous model prediction $\hat{y}_{t-1}$ with probability $\epsilon$ or the previous ground truth prediction $y_{t-1}$ with probability $1 - \epsilon$. The model prediction $\hat{y}_{t-1}$ is obtained by sampling a token according to the probability distribution by $P(y_{t-1}|h_{t-1})$. At the beginning of training, the sampled token can be very random. Thus, the probability $\epsilon$ is set very low initially and increased over time.

One major limitation of scheduled sampling is that at each time step, the target sequences can be incorrect since they are randomly selected from the ground truth data or model predictions, regardless of how input was chosen (Ranzato et al., 2015). Thus, we use reinforcement learning techniques to further optimize models on translation metrics directly.

### 2.2 Reinforcement Learning

Following Ranzato et al. (2015) and Rennie et al. (2017), we use REINFORCE with baseline to directly optimize the evaluation metric.

According to the reinforcement learning literature (Sutton et al., 1998), the neural network, $\theta$, defines a policy $p_\theta$, that results in an "action" that is the prediction of next word. After generating the end-of-sequence term (EOS), the model will get a reward $r$, which can be the evaluation metric, e.g. BLEU score, between the golden and generated sequence. The goal of training is to minimize the negative expected reward.

$$L(\theta) = -\mathbb{E}_{w^s \sim p_\theta} \left[ r(w^s) \right]. \quad (2)$$

where sentence $w^s = (w_1^s, ..., w_T^s)$.

In order to compute the gradient $\nabla_\theta L(\theta)$, we use the REINFORCE algorithm, which is based on the observation that the expected gradient of a non-differentiable reward function can be computed as follows:

$$\nabla_\theta L(\theta) = -\mathbb{E}_{w^s \sim p_\theta} \left[ r(w^s) \nabla_\theta \log p_\theta(w^s) \right]. \quad (3)$$

The policy gradient can be generalized to compute the reward associated with an action value *relative* to a reference reward or *baseline* $b$:

$$\nabla_\theta L(\theta) = -\mathbb{E}_{w^s \sim p_\theta} \left[ (r(w^s) - b) \nabla_\theta \log p_\theta(w^s) \right]. \quad (4)$$

The baseline does not change the expected gradient, but importantly, it can reduce the variance of the gradient estimate. We use the baseline introduced in Rennie et al. (2017) which is obtained by the current model with greedy decoding at test time.

$$b = r(\hat{w^s}) \quad (5)$$

where $\hat{w^s}$ is generated by greedy decoding.

For each training case, we approximate the expected gradient with a single sample $w^s \sim p_\theta$:

$$\nabla_\theta L(\theta) \approx -(r(w^s) - b) \nabla_\theta \log p_\theta(w^s). \quad (6)$$

|  | Train | Dev. | Vocab. | Vocab. after BPE |
|---|---|---|---|---|
| En | 2,900 | 1,014 | 10,212 | 7,633 |
| De | 2,900 | 1,014 | 18,726 | 5,942 |
| Fr | 2,900 | 1,014 | 11,223 | 6,457 |
| Cs | 2,900 | 1,014 | 22,400 | 8,459 |

Table 1: Statistics of Flickr30K Dataset

|  | En-De | En-Fr | En-Cs |
|---|---|---|---|
| NMT | 39.64 | 58.36 | 31.27 |
| NMT+SS | 40.19 | 58.67 | 31.38 |
| NMT+SS+RL | 40.60 | 58.80 | 31.73 |
| MNMT | 39.27 | 57.92 | 30.84 |
| MNMT+SS | 39.87 | 58.80 | 31.21 |
| MNMT+SS+RL | 40.39 | 58.78 | 31.36 |
| NMT Ensemble | **42.54** | 61.43 | **33.15** |
| MIX Ensemble | 42.45 | **61.45** | 33.11 |

Table 2: BLEU scores of different approaches on the validation set. Details of the ensemble models are described in Table 9.

### 2.3 Ensembling

In our experiments with relatively small training dataset, the translation qualities of models with different initializations can vary notably. To make the performance much more stable and improve the translation quality, we ensemble different models during decoding to achieve better translation.

To ensemble, we take the average of all model outputs:

$$\hat{y_t} = \sum_{i=1}^{N} \frac{\hat{y_t^i}}{N} \tag{7}$$

where $\hat{y_t^i}$ denotes the output distribution of $i$th model at position $t$. Similar to Zhou et al. (2017), we can ensemble models trained with different architectures and training algorithms.

## 3 Experiments

### 3.1 Datasets

We perform experiments using Flickr30K (Elliott et al., 2016) which are provided by the WMT organization. Task 1 (Multimodal Machine Translation) consists of translating an image with an English caption into German, French and Czech. Task 1b (Multisource Multimodal Machine Translation) involves translating parallel English, German and French sentences with accompanying image into Czech.

As shown in Table 1, both tasks have 2900 training and 1014 validation examples. For preprocessing, we convert all of the sentences to lower case, normalize the punctuation, and tokenize. We employ byte-pair encoding (BPE) (Sennrich et al., 2015) on the whole training data including the four languages and reduce the source and target language vocabulary sizes to 20k in total.

### 3.2 Training details

The image feature is extracted using ResNet-101 (He et al., 2016) convolutional neural network trained on the ImageNet dataset. Our implementation is adapted from Pytorch-based OpenNMT (Klein et al., 2017). We use two layered bi-LSTM (Sutskever et al., 2014) as the encoder and share the vocabulary between the encoder and the decoder. We adopt length reward (Huang et al., 2017) on En-Cs task to find the optimal sentence length. We use a batch size of 50, SGD optimization, dropout rate as 0.1 and learning rate as 1.0. Our word embeddings are randomly initialized of dimension 500.

To train the model with scheduled sampling, we first set probability $\epsilon$ as 0, and then gradually increase it 0.05 every 5 epochs until it's 0.25. The reinforcement learning models are trained based on those models pre-trained by scheduled sampling.

### 3.3 Results for task 1

To study the performance of different approaches, we conduct an ablation study. Table 2 shows the BLEU scores on validation set with different models and training methods. Generally, models with scheduled sampling perform better than baseline models, and reinforcement learning further improves the performance. Ensemble models lead to substantial improvements over the best single model by about +2 to +3 BLEU scores. However, by including image information, MNMT per-

| Task | System | NMT+SS | NMT+SS+RL | MNMT+SS | MNMT+SS+RL |
|---|---|---|---|---|---|
| En-De | NMT | 7 | 6 | 0 | 0 |
|  | MIX | 7 | 6 | 5 | 4 |
| En-Fr | NMT | 9 | 5 | 0 | 0 |
|  | MIX | 9 | 0 | 3 | 0 |
| En-Cs | NMT | 7 | 6 | 0 | 0 |
|  | MIX | 7 | 6 | 5 | 4 |

Table 3: Number of different models used for ensembling.

| | Rank | BLEU | METEOR | TER |
|---|---|---|---|---|
| OSU-BD-NMT | 1 | **32.3** | 50.9 | 49.9 |
| OSU-BD-MIX | 2 | 32.1 | 50.7 | **49.6** |
| LIUMCVC-MNMT-E | 3 | 31.4 | 51.4 | 52.1 |
| UMONS-DeepGru | 4 | 31.1 | **51.6** | 53.4 |
| LIUMCVC-NMT-E | 5 | 31.1 | 51.5 | 52.6 |
| SHEF1-ENMT | 6 | 30.9 | 50.7 | 52.4 |
| Baseline | - | 27.6 | 47.4 | 55.2 |

Table 4: En-De results on test set. 17 systems in total. (Only including constrained models).

| | Rank | BLEU | METEOR | TER |
|---|---|---|---|---|
| LIUMCVC-MNMT-E | 1 | **39.5** | 59.9 | 41.7 |
| UMONS | 2 | 39.2 | **60** | 41.8 |
| LIUMCVC-NMT-E | 3 | 39.1 | 59.8 | 41.9 |
| OSU-BD-NMT | 4 | 39.0 | 59.5 | **41.2** |
| SHEF-MLT | 5 | 38.9 | 59.8 | 41.5 |
| OSU-BD-MIX | 9 | 38.6 | 59.3 | 41.5 |
| Baseline | - | 28.6 | 52.2 | 58.8 |

Table 5: En-Fr results on test set. 14 systems in total. (Only including constrained models).

| | Rank | BLEU | METEOR | TER |
|---|---|---|---|---|
| OSU-BD-NMT | 1 | **30.2** | 29.5 | **50.7** |
| OSU-BD-MIX | 2 | 30.1 | **29.7** | 51.2 |
| SHEF1-ENMT | 3 | 29.0 | 29.4 | 51.1 |
| SHEF-LT | 4 | 28.3 | 29.1 | 51.7 |
| SHEF-MLT | 5 | 28.2 | 29.1 | 51.7 |
| SHEF1-MFS | 6 | 27.8 | 29.2 | 52.4 |
| Baseline | - | 26.5 | 27.7 | 54.4 |

Table 6: En-Cs results on test set. 8 systems in total. (Only including constrained models).

| | En-Cs | Fr-Cs | De-Cs | (En+Fr+De)-Cs |
|---|---|---|---|---|
| NMT | **31.27** | 28.48 | 26.96 | 29.47 |
| MNMT | 30.84 | 27.02 | 25.99 | 29.23 |

Table 7: BLEU scores on validation set for task 1B

forms better than NMT only on the En-Fr task with scheduled sampling.

Table 4, 5 and 6 show the test set performance of our models on En-De, En-Fr and En-Cs subtasks with other top performance models. We rank those models according to BLEU. Our submitted systems rank first in BLEU and TER on En-De and En-Cs subtasks.

### 3.4 Results for task 1B

Table 7 shows the results on validation set without sequence training. En-Cs, Fr-Cs, De-Cs are models trained from one language to another. (En+Fr+De)-Cs models are trained using multiple source data. Similar to the Shuffle method discussed in multi-reference training (Zheng et al., 2018), we randomly shuffle the source data in all languages and train using a traditional attention based-neural machine translation model in every epoch. Since we do BPE on the whole training data, we can share the vocabulary of different languages during training. The results show that models trained using single English to Czech data perform much better than the rest.

| | Rank | BLEU | METEOR | TER |
|---|---|---|---|---|
| OSU-BD-NMT | 1 | **26.4** | 28.0 | **52.1** |
| OSU-BD-MIX | 1 | **26.4** | **28.2** | 52.7 |
| SHEF1-ARNN | 3 | 25.2 | 27.5 | 53.9 |
| SHEF-CON | 4 | 24.7 | 27.6 | **52.1** |
| SHEF-MLTC | 5 | 24.5 | 27.5 | 52.5 |
| SHEF1-ARF | 6 | 24.1 | 27.1 | 54.6 |
| Baseline | - | 23.6 | 26.8 | 54.2 |

Table 8: Task 1B multi-source translation results on test set. 6 systems in total.

| Task | System | Model Rank | | | | Team Rank | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Num $^{\dagger}$ | BLEU | MET. | TER | Num $^{\ddagger}$ | BLEU | MET. | TER |
| En-De | NMT | 11 | 1 | 4 | 2 | 5 | 1 | 3 | 1 |
| | MIX | 11 | 2 | 5 | 1 | | | | |
| En-Fr | NMT | 11 | 4 | 9 | 1 | 6 | 3 | 5 | 1 |
| | MIX | 11 | 9 | 10 | 3 | | | | |
| En-Cs | NMT | 6 | 1 | 1 | 1 | 3 | 1 | 1 | 1 |
| | MIX | 6 | 2 | 2 | 3 | | | | |
| En-Cs (1B) | NMT | 6 | 1 | 2 | 1 | 3 | 1 | 1 | 1 |
| | MIX | 6 | 1 | 1 | 5 | | | | |

Table 9: Rank of our models. $^{\dagger}$ represents the total number of models. $^{\ddagger}$ represents the total number of teams.

Table 8 shows results on test set. The submitted systems are the same as those used in En-Cs task of task 1. Although we only consider the English source during training, our proposed systems still rank first among all the submissions.

## 4 Conclusions

We describe our systems submitted to the shared WMT 2018 multimodal translation tasks. We use sequence training methods which lead to substantial improvements over strong baselines. Our ensembled models achieve the best performance in BLEU score for three subtasks: En-De, En-Cs of task 1 and (En+De+Fr)-Cs task 1B.

## Acknowledgments

This work was supported in part by DARPA grant N66001-17-2-4030, and NSF grants IIS-1817231 and IIS-1656051. We thank the anonymous reviewers for suggestions and Juneki Hong for proofreading.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179.

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost van de Weijer. 2017. Lium-cvc submissions for wmt17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation*, pages 432–439.

Iacer Calixto and Qun Liu. 2017. Incorporating global visual features into attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 992–1003.

D. Elliott, S. Frank, K. Sima'an, and L. Specia. 2016. Multi30k: Multilingual english-german image descriptions. *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation*, pages 215–233.

Ekaterina Garmash and Christof Monz. 2016. Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1409–1418.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *Conference on Computer Vision and Pattern Recognition CVPR*.

Liang Huang, Kai Zhao, and Mingbo Ma. 2017. When to finish? optimal beam search for neural text generation (modulo beam size). In *EMNLP 2017*.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *ArXiv e-prints*.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Improved image captioning via policy gradient optimization of spider. In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3, page 3.

Mingbo Ma, Dapeng Li, Kai Zhao, and Liang Huang. 2017. Osu multimodal machine translation system report. In *Proceedings of the Second Conference on Machine Translation*, pages 465–469.

Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. 2016. Classify or select: Neural architectures for extractive document summarization. *CoRR*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *CVPR*, volume 1, page 3.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *Proceedings of the 27th International Conference on Neural Information Processing Systems*.

Richard S Sutton, Andrew G Barto, et al. 1998. *Reinforcement learning: An introduction*. MIT press.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence - video to text. In *ICCV*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*.

Renjie Zheng, Mingbo Ma, and Huang Liang. 2018. Multi-reference training with pseudo-references for neural translation and text generation. *arXiv preprint arXiv:1808.09564*.

Long Zhou, Wenpeng Hu, Jiajun Zhang, and Chengqing Zong. 2017. Neural system combination for machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 378–384.

# Translation of Biomedical Documents with Focus on Spanish-English

**Mirela-Stefania Duma and Wolfgang Menzel**
University of Hamburg
Natural Language Systems Division
{mduma, menzel}@informatik.uni-hamburg.de

## Abstract

For the WMT 2018 shared task of translating documents pertaining to the Biomedical domain, we developed a scoring formula that uses an unsophisticated and effective method of weighting term frequencies and was integrated in a data selection pipeline. The method was applied on five language pairs and it performed best on Portuguese-English, where a BLEU score of 41.84 placed it third out of seven runs submitted by three institutions. In this paper, we describe our method and results with a special focus on Spanish-English where we compare it against a state-of-the-art method. Our contribution to the task lies in introducing a fast, unsupervised method for selecting domain-specific data for training models which obtain good results using only 10% of the general domain data.

## 1 Introduction

The 2018 Biomedical Translation Task, held as part of the Third Conference on Machine Translation, aims at evaluating systems on scientific publications from Medline (Neves et al., 2018). The task is particularly challenging as there is still not enough bilingual medical data available for training high quality Machine Translation (MT) systems. We develop and apply a data selection method on five out of the nine language pairs addressed by the task: English-Spanish, Spanish-English, English-Portuguese, Portuguese-English and English-Romanian.

Data selection, as a domain adaptation technique, exploits all available (bilingual) general domain corpora with the purpose of extracting sentences that have a strong relationship to a given in-domain. All sentences from the general domain pool are scored according to a similarity function/ algorithm/ method and after being sorted, the most similar ones are selected to take part in the MT training pipeline. The subsampling is usually done using a threshold, which is the number of sentence (pairs) or a percentage of the sentences to be considered in-domain.

We introduce a data selection method which is fast to apply and yields good results when compared with a strong baseline and a state-of-the-art method. The simplicity of the method has at its core term frequencies and a newly developed similarity function. On the one hand, no models need to be trained and the method is unsupervised, but on the other hand, the method does not consider the context of the words or their semantics. However, the results are very encouraging with BLEU (Papineni et al., 2002) scores between 31.05 and 41.84 for four language pairs.

The paper is structured as follows: the next section briefly presents related work, Section 3 describes the experimental results along with a description of our algorithm, Section 4 gives an overview of the results obtained in the task and additional experiments and the last section presents conclusions and future work.

## 2 Related work

Related work in data selection is ample, therefore this section only mentions methods that fit in the same category with our method and we also shortly describe the widely known state-of-the art method of performing data selection, introduced by Axelrod et al. (2011), since it is the chosen method for comparing results in this paper.

Our scoring function relies heavily on term frequency. Therefore, it falls in the category of TF-IDF[1] based approaches. Hildebrand et al. (2005) uses TF-IDF to produce vector representations of sentences. Then the cosine of the angle between the sentence vectors is interpreted as the similar-

[1] Term Frequency - Inverse Document Frequency

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 637–643
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64071

ity between the sentences. A similar approach is given in Eck et al. (2005) where a weighting scheme based on TF-IDF by means of unseen n-grams and sentence length is applied and cosine is also used as means of determining sentence similarities. In contrast to these methods, we use only the term frequency in computing our similarity scores and we make no use of the cosine. Instead, we focus on the relative difference between a term that appears in the general domain and in the in-domain and simply multiply it by a weighting scheme that has empirically proved to be effective. Our method is also related to the other methods from the TF-IDF category with respect to its simplicity.

To compare our results with other approaches we apply the modified Moore-Lewis method which is based on (Moore and Lewis, 2010): given the source side of an in-domain corpus and a random subsample of the source side of a general domain corpus, a language model (LM) is trained on each one of them. The sentences from the general domain are scored by the difference of the cross-entropy of a sentence according to the in-domain LM and the cross-entropy of the same sentence according to the general domain LM. Axelrod et al. (2011) modified the scoring by applying the same procedure also to the target side of the corpora and afterwards summing the scores. We refer to this method as *MML (modified Moore-Lewis)* in the rest of the paper.

## 3 Experiments

This section describes the experimental settings including the corpora and the tools used, as well as the data selection algorithm we developed.

### 3.1 Corpora

The general domain data consisted of a concatenation of the Commoncrawl[2] corpora and the Wikipedia (Wolk and Marasek, 2014) corpora for English-Spanish and Spanish-English, Paracrawl[3] and Wikipedia for English-Portuguese and Portuguese-English and Paracrawl for English-Romanian. For the in-domain, we used the EMEA (Tiedemann, 2012) corpora for all language pairs and the Scielo corpora (health and biological) provided by the WMT 2016 Biomedical task (Neves et al., 2016) for all language pairs except for English-Romanian where Scielo training data was not available.

The development set for the English-Spanish and Spanish-English experiments was a concatenation of the Khreshmoi development set from the Medical Task of WMT 2014[4] and the ECDC corpus made available by UFAL[5]. The motivation for using a concatenation of two medical development sets is that we aimed at diversity in the medical data. Even though ECDC is a very small corpus consisting of only 2357 sentence pairs (for English-Spanish), combining it with Khreshmoi (500 sentence pairs) would have resulted in a quite big development set which would have made the tuning of the SMT systems very time and memory intensive. Therefore, we applied a cleaning step to ECDC which meant limiting the size of the sentences to a minimum of 20 words and a maximum of 80 words. After applying this preprocessing step, the ECDC set was down to 850 sentences, resulting in a total development set of 1350 sentences. For the experiments involving Portuguese, a sample of 1000 sentences from the Scielo development set from WMT 2016[6] was used for tuning purposes. As for the Romanian experiments, also a sample of 1000 sentences was used, but from the ECDC corpus.

Statistics including the number of sentences after preprocessing for every corpus used for the training of the MT systems is given in Table 1.

| Track / Corpora | EN-ES | EN-PT | EN-RO |
|---|---|---|---|
| Commoncrawl | 1.8M | - | - |
| Paracrawl | - | 2.1M | 2.4M |
| Wikipedia | 1.6M | 1.6M | - |
| EMEA | 678K | 1.08M | 994K |
| Scielo-gma 2016 | 166K | 613K | - |

Table 1: Corpora used for DSTF

### 3.2 Tools

For text processing we used the *nltk* toolkit(Bird et al., 2009), the WordNet (Fellbaum, 1998) lemmatizer for English and the Snowball stemmer (F. Porter, 2001) for Spanish, Portuguese and Romanian.

The SMT systems were trained using the Moses toolkit (Koehn et al., 2007) and the Experiment Management System (Koehn, 2010). The preprocessing of the data consisted in tokenization,

[2] http://commoncrawl.org/

[3] https://paracrawl.eu/index.html

[4] http://www.statmt.org/wmt14/medical-task/

[5] http://ufal.mff.cuni.cz/ufal_medical_corpus

[6] http://www.statmt.org/wmt16/biomedical-translation-task.html

**Algorithm 1** DSTF Filtering

```
procedure PREPROCESS_CORPUS($\mathcal{C}$)
    $tokenize(\mathcal{C})$
    $lowercase(\mathcal{C})$
    $removeStopWords(\mathcal{C})$
    $lemmatize(\mathcal{C})$                                   ▷ or stem if unavailable
    $keepWords(\mathcal{C})$
    $wordCount(\mathcal{C})$
procedure FILTER($\mathcal{GEN}_{side}, \mathcal{IN}_{side}$)         ▷ side refers to either source or target
    Preprocess_Corpus($\mathcal{GEN}_{side}$)
    Preprocess_Corpus($\mathcal{IN}_{side}$)
    for each sentence $s \in \mathcal{GEN}_{side}$ do
        for each word $w \in s$ do
            if $count(w, \mathcal{GEN}_{side}) = 0$ then
                $weight = 0$
            else
                $weight = count(w, \mathcal{IN}_{side}) / count(w, \mathcal{GEN}_{side})$
            $score_w = \left( \frac{2 \cdot (count(w, \mathcal{IN}_{side}) - count(w, \mathcal{GEN}_{side}))}{count(w, \mathcal{IN}_{side}) + count(w, \mathcal{GEN}_{side})} \right)^2 \cdot weight$
        $score_s \mathrel{+}= score_w$                ▷ all intermediate scores contribute to the final score
```

cleaning, lowercasing and normalizing punctuation. Our language model (LM) was obtained by interpolating (Schwenk and Koehn, 2008) the LM estimated using the general domain data and the LM estimated on the in-domain data. We used the SRILM toolkit (Stolcke, 2002) and Kneser-Ney discounting (Kneser and Ney, 1995) for estimating 5-grams LMs. All the experiments benefited from the interpolated language model, including the strong baseline and the *MML* experiment. As for the chosen state-of-the-art method, *MML*, we used the implementation available from Moses.

Tuning of the systems was done with MERT (Och, 2003) and GIZA++ (Och and Ney, 2003) using the default *grow-diag-final-and* alignment symmetrization method for word alignment.

### 3.3 Data selection using Term Frequency

Using bag of words to represent sentences and term frequency to compute similarity became unpopular due to its limitations, namely no integration of semantic information and ignoring the context of words (Le and Mikolov, 2014). However, through the work presented here we aim at applying this straightforward method to data selection for SMT with a new weighting scheme. Our scoring algorithm builds a profile consisting of word frequencies for each domain, for the source language and the target language. To build the profile for a corpus, all of its sentences undergo a preprocessing step: tokenization, lowercasing, removal of stop words and lemmatization or stemming in the case a lemmatizer was not available for a language (procedure $Preprocess\_Corpus$). In the end, numbers or punctuation marks are ignored and only words contribute to the scoring. For word count occurrence we used the script $ngram - count$ from SRILM.

Algorithm 1 can be applied either on the source or on the target sides of the corpora. For example, when considering the source side, for every sentence from the lemmatized (or stemmed) general domain data, we iterate through all its words. Given sentence $s$ and the word $w$, we square the relative difference between the term frequency of $w$ in the in-domain profile, $count(w, \mathcal{IN}_{side})$, and the term frequency of $w$ in the general domain profile, $count(w, \mathcal{GEN}_{side})$. We use the same relative difference formula as in (Kešelj et al., 2003) which uses character n-grams and profiles built using the most frequent character ngrams for authorship attribution. In contrast to this, we used all the words appearing in the corpora and modified the formula by introducing a weighting scheme. Note that due to the squaring, the direction of the subtraction does not matter. The difference is multiplied by a weight and the arithmetic mean of $count(w, \mathcal{IN}_{side})$ and $count(w, \mathcal{GEN}_{side})$. The weight represents the impact that $w$ made in the sentence and we empirically determined it. When using only the formula from Kešelj et al. (2003) adapted to our data selection task, the results are of poor quality. Our contribution to the formula lies in introducing the weighting scheme which gives much better results than the original formula. To profit from both the source and the target corpora, summing up the scores for the source language and the scores for the target language seems to be an attractive solution. We refer to our method as *DSTF (Data Selection via Term Frequency)*.

The method has a very important advantage if compared to state-of-the-art methods: scoring is very fast for a general domain corpus (on average, the scoring step took half an hour). The results are satisfactory and will be presented in the following section.

## 4 Results

We report the automatic evaluation results obtained in the WMT task for five language pairs and then we present further experiments for the Spanish-English language pair. BLEU was used as an evaluation metric by the WMT Biomedical organizers and in addition to BLEU we also used METEOR (Lavie and Agarwal, 2005) for further evaluating the Spanish-English experiments.

### 4.1 WMT Biomedical Results

Each team was allowed to submit a maximum of three runs. For every language pair that we used to evaluate our method on, we submitted three runs as follows: the first run only considers the scores obtained using the English side of the training corpora, the second run made use of only the non-English side of the training corpora and for the third run the scores for both the source and the target sides were summed up to form a single score.

The aim of data selection is to identify in the general domain pool the top $\mathcal{N}$ most similar sentences to an in-domain, where $\mathcal{N}$ is determined empirically and is usually a small number or percentage. We experimented for this paper with $\mathcal{N} = 10\%$ since the maximum of runs allowed was three and we had three variations of the method, but we intend to conduct a range of experiments with more percentage values in future work. Table 2 presents the number of sentence pairs that were subsampled along with the total number of sentence pairs that were used in the training of MT systems.

| Language pair | EN-ES | EN-PT | EN-RO |
|---|---|---|---|
| 10% of Gen | 350K | 378K | 245K |
| total training data | 1.62M | 2.07M | 1.24M |

Table 2: Corpora used for DSTF

The BLEU results obtained using *DSTF* are encouraging: a BLEU score of 41.84 for Portuguese-English ranked our method on the third place out of seven runs submitted by three institutions. For English-Portuguese, our BLEU scores are close to 34 for all runs. The Spanish-English automatic evaluation achieved scores around 35-36 and for English-Spanish around 31. The smallest BLEU scores were measured for English-Romanian where we obtained scores close to 14. This is not surprising considering the fact that compared to the other language pairs there was less biomedical training data available. In particular, no Scielo training corpus was available although translating from English to a morphologically rich language like Romanian is considered difficult. The BLEU scores for each run are given in Table 3. We note that the differences between each run, for every language pair, are insignificant except for one language pair, therefore we conclude that either one of the algorithm variations can be successfully applied as a fast data selection technique that yields good translations (BLEU scores between 31 and 42 for four out of five language pairs).

| Language pair | EN-ES | ES-EN | EN-PT | PT-EN | EN-RO |
|---|---|---|---|---|---|
| run 1 | 31.32 | 36.16 | 34.92 | 41.84 | 14.60 |
| run 2 | 31.05 | 35.17 | 34.19 | 41.80 | 14.39 |
| run 3 | 31.33 | 36.05 | 34.49 | 41.79 | 14.07 |

Table 3: BLEU scores reported by WMT

### 4.2 Spanish-English Additional Experiments

For Spanish-English, the best performing variant of our method was run 1 - using only the English side of the corpora in the algorithm. We evaluated our *DSTF-EN* method against a strong baseline (that uses an interpolated LM), a baseline trained using only the in-domain data and the state-of-the-art method *MML* for the Spanish-English language pair[7]. Following recommendations from H. Clark et al. (2011) and standard practices, we tuned the systems three times and report in Table 4 the averaged BLEU scores.

[7] Due to time limitations, we will evaluate further language pairs against *MML* in the future work.

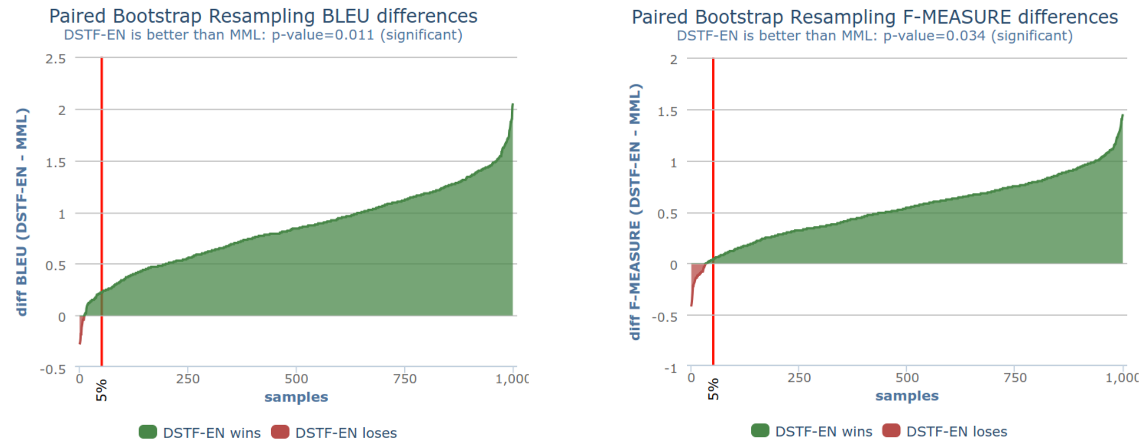


Figure 1: Paired bootstrap resampling graphs using BLEU differences between *DSTF-EN* and *MML* (left graph) and using F-measure differences (right graph)

| System | BS-strong | BS-IN | MML | DSTF-EN |
|---|---|---|---|---|
| BLEU | 34.96 | 32.44 | 34.62 | 35.40 |
| METEOR | 35.56 | 34.51 | 35.42 | 35.54 |

Table 4: averaged BLEU scores for Spanish-English

According to the BLEU scores, our method outperformed both baselines and gained almost 1 BLEU point over *MML*. The strong baseline is very competitive with both data selection methods. This can easily be explained, since the system relies on the same interpolated language model as *DSTF-EN* and *MML*. There is a 3 BLEU points difference between our results and the baseline trained only the in-domain data and almost half a point BLEU score difference between the strong baseline and our method. With respect to the METEOR scores, our method again outperforms the state-of-the-art approach.

In order to determine whether our method (*DSTF-EN*) outperforms the state-of-the-art method (*MML*) from a statistical point of view, we applied paired bootstrap resampling (Koehn, 2004). The MTCompar-Eval tool (Klejch et al., 2015; Sudarikov et al., 2016) was used for this purpose where the source, reference and one or more system translations are used in the analysis. For our analysis we selected the best translation of each system according to their BLEU scores[8].

Figure 1 depicts the paired bootstrap resampling BLEU graph (left side) and the F-measure graph (right side). The x-axis is represented by 1000 resamples of the test set and the y-axis represents the difference in BLEU (respectively F-measure) between *DSTF-EN* and *MML* for all resamples. The p-value from the first graph in Figure 1 reports that in 11 cases out of the 1000 resamples, the state-of-the-art method performed better in terms of BLEU than our method (marked with a small red area in the graph). A similar behaviour can be oserved in the right graph from Figure 1 where in 34 cases out of 1000, *MML* outperformed *DSTF-EN* in terms of F-measure. Therefore in 96.6% of the times our method wins over the state-of-the-art when using the F-measure and in 98.9% of the cases, our method is better than *MML* when evaluating with BLEU (large green areas in the graphs). We conclude that our method has a statistical significant performance in comparison with the state-of-the-art method when selecting the 10% of the general domain sentences that were most similar to the in-domain.

## 5 Conclusions and Future Work

We introduced an unsophisticated data selection method based on word frequencies which scores general domain corpora in half an hour (on average when considering all general corpora for five language pairs). Our method yields good results in the WMT task, as well as in comparison with a state-of-the-art method and a strong baseline (for Spanish-English). Further analysis and experiments will be carried out in future work to assess whether the improvement of our method over the state-of-the-art that we observed for Spanish-English is also statistically significant for other language pairs.

[8] We tuned three times and averaged the BLEU scores

## References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain Adaptation via Pseudo In-domain Data Selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Stroudsburg, PA, USA. Association for Computational Linguistics.

Steven Bird, Edward Loper, and Ewan Klein. 2009. Natural Language Processing with Python. In *O'Reilly Media Inc.*

Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Low Cost Portability for Statistical Machine Translation based on N-gram Frequency and TF-IDF. In *International Workshop on Spoken Language Translation, IWSLT 2005, Pittsburgh, PA, USA*, pages 61–67.

M F. Porter. 2001. Snowball: A language for stemming algorithms. In *Retrieved March*, volume 1.

Christiane Fellbaum. 1998. WordNet: An Electronic Lexical Database. In *Cambridge, MA: MIT Press*.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 176–181.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval. In *Proceedings of EAMT*, pages 133–142.

Vlado Kešelj, Fuchun Peng, Nick Cercone, and Calvin Thomas. 2003. N-Gram-Based Author Profiles For Authorship Attribution. In *Proceedings of the Conference Pacific Association for Computational Linguistics PACLING 2003*.

Ondrej Klejch, Eleftherios Avramidis, Aljoscha Burchardt, and Martin Popel. 2015. MT-ComparEval : Graphical Evaluation Interface for Machine Translation Development. In *The Prague Bulletin of Mathematical Linguistics, Number 104*, pages 63–74.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for N-gram language modeling. In *Proceedings ICASSP*, pages 181–184.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *EMNLP*.

Philipp Koehn. 2010. An Experimental Management System. *Prague Bull. Math. Linguistics*, 94:87–96.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alon Lavie and Abhaya Agarwal. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 65–72.

Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org.

Robert C. Moore and William Lewis. 2010. Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 220–224, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mariana Neves, Antonio Jimeno Yepes, Aurelie Névéol, Cristian Grozea, Amy Siu, Madeleine Kittner, and Karin Verspoor. 2018. Findings of the WMT 2018 Biomedical Translation Shared Task. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Mariana Neves, Antonio Jimeno Yepes, and Aurélie Névéol. 2016. The Scielo Corpus: a Parallel Corpus of Scientific Publications for Biomedicine. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Holger Schwenk and Philipp Koehn. 2008. Large and diverse language models for statistical machine translation. In *In Proceedings of The Third International Joint Conference on Natural Language Processing (IJCNP*.

Andreas Stolcke. 2002. SRILM-an extensible language modeling toolkit. In *Interspeech*, volume 2002.

Roman Sudarikov, Martin Popel, Ondrej Bojar, Aljoscha Burchardt, and Ondrej Klejch. 2016. Using MT-ComparEval. In *Proceedings of the LREC 2016 Workshop "Translation Evaluation – From Fragmented Tools and Data Sets to an Integrated Ecosystem"*.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Krzysztof Wolk and Krzysztof Marasek. 2014. Building Subject-aligned Comparable Corpora and Mining it for Truly Parallel Sentence Pairs. In *Procedia Technology, 18*, pages 126 – 132. Elsevier.

# Ensemble of Translators with Automatic Selection of the Best Translation – the submission of FOKUS to the WMT 18 biomedical translation task –

**Cristian Grozea**
Fraunhofer FOKUS
cristian.grozea@fokus.fraunhofer.de

## Abstract

This paper describes the system of Fraunhofer FOKUS for the WMT 2018 biomedical translation task. Our approach, described here, was to automatically select the most promising translation from a set of candidates produced with NMT (Transformer) models. We selected the highest fidelity translation of each sentence by using a dictionary, stemming and a set of heuristics. Our method is simple, can use any machine translators, and requires no further training in addition to that already employed to build the NMT models. The downside is that the score did not increase over the best in ensemble, but was quite close to it (difference about 0.5 BLEU).

## 1 Introduction

As previously noted in (Sennrich et al., 2016; Zhou et al., 2017), the neural machine translation models tend to provide good fluency but sometimes at the expense of the fidelity – they may struggle to cope with rare words, and can exhibit poor coverage/fidelity by ignoring altogether parts of the source.

By training even the same networks on different data one obtains models that have different strengths and weaknesses, sometimes one model provides the better translation, sometimes another one, even if on average they are of rather equal performance.

Our approach, described here, was to automatically select the best translation from a set of candidates produced by an ensemble of neural translators. As the fluency was generally good, as is typically the case with NMT, our heuristic scoring of the translation quality focused on the bidirectional coverage, estimated by making use of a dictionary aided by a set of heuristic rules for the words not found in the dictionary. We aimed to select thus automatically the highest fidelity translation.

| Name | Description | Pairs |
|---|---|---|
| MED | medication accompanying patient information leaflets from the UFAL Medical Corpus 1.0(ufa) En-Ro(subset) | 1048757 |
| NEWS | SE Times En-Ro + Europarl 2017 En-Ro | 612422 |

Table 1: Datasets used to train and validate the neural networks

Combining translators is not new, the most interesting result known to us is (Zhou et al., 2017), where the authors report improvements of over 5 BLEU points in Chinese-to-English translation by combining the outputs of SMT and NMT systems using a neural network.

Our method is much simpler, has the additional advantage of using the NMT models as blackboxes, and requires no further training in addition to that already employed to build the NMT models. The downside is that the BLEU score did not increase over the best in the ensemble (was within 0.5 BLEU of it) on a non directly comparable task, the biomedical field English-to-Romanian translation task of the WMT 2018 workshop.

## 2 Methods

The datasets listed in Table 1 have been used for training and validation in various ways. We have grouped the En-Ro parallel corpora available to us in two groups, Medical (short: MED) and News+ EU Parliament debates (short: NEWS).

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 644–647
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64072

| Letter | MED | NEWS |
|---|---|---|
| Incorrect ş (unicode 351) | 273258 | 289092 |
| Incorrect ţ (unicode 355) | 474633 | 323086 |
| Correct ș (unicode 537) | 28434 | 101095 |
| Correct ț (unicode 539) | 48896 | 109172 |

Table 2: Diacritics usage in the datasets used here – number of lines containing a certain letter

### 2.1 Considerations specific to the Romanian language concerning the character codes used for diacritics

The Romanian language uses 5 letters with diacritics: ă, â, î, ș, ț. Before a 2003 decision of the Romanian Academy, other characters were in wide use instead of ș (unicode 537) and ț (unicode 539): cedilla-based ş (unicode 351) and ţ (unicode 355). The history of decades of broken support in various operating systems and character sets is related at http://kitblog.com/2008/10/romanian_diacritic_marks.html. The diacritics in Romanian are fairly redundant, automatic restoration is possible, with less than 1% errors (Grozea, 2012). The changes over the years, starting with using no diacritics at all in the 1980s and early 1990s, then using cedilla based ones, then comma based ones led to heterogeneous corpora used in NLP: some texts have no diacritics at all, some have the wrong diacritics, some have a mixture of wrong and correct diacritics. This affects multiple NLP tasks, including translation. Learning from examples to translate into Romanian is more difficult than it should be when the examples sampled from various corpora alternate randomly the diacritics they use. The diacritics usage statistics for the datasets used here is given in Table 2.

### 2.2 NMT models

We have used for our experiments the tensor2tensor (T2T) implementation of the Transformer network (Vaswani et al., 2018). Several training runs have been performed, described in Table 3. The training has been interrupted manually when the loss on the validation set started to increase (early stop), as judged by the experimenter monitoring the evolution of the loss on tensorboard. As such, small fluctuations of the loss do not lead to a too early stop.

The external BPE preprocessing was performed using scripts from the SMT system Moses (Koehn et al., 2007).

### 2.3 Ensemble Aggregation by Translation Selection

Each model has been used to translate all source sentences from English to Romanian. The aggregation of those outputs has been performed by selecting automatically the translation having the highest quality.

In order to assess the quality of the sentence translations we have computed the percentage of words in the source that have a correspondent in the translation (coverage) and the percentage of the words in the translation that have a correspondent in the source. The minimum of those two numbers between 0 and 1 is taken as the quality of the translation. Once a correspondent is found, it it removed from the next searches (in a greedy fashion, as opposed to the alternative of maximizing the matching with dynamic programming). A word matching is evaluated to 1, when the pair is found in the dictionary, after stemming and the normalization described below, that is applied to the dictionary as well. A pair of words that become identical after stemming and normalization lead to a matching of value 0.3. If the words normalized after stemming are not identical, not too short (they are at least 4 characters) and one of them is a prefix of the other, then the matching is evaluated to 0.2. When computing the coverage mentioned above, the sum of the word pair matching quality is divided by the total number of words.

The preprocessing steps for text normalization, applied both to the sentence pair (source and translation) and on the dictionary are:

- Diacritics removal;
- Replacing of *ph* with *f*, of *y* with *i* and of *ff* with *f*.

The aim of the diacritics removal was to cope with the heterogeneous codes for the letters with diacritics and to cover also for the texts without diacritics. The aim of the substitution of the groups of letters was to increase the chance to recognize proper translation of medical terms originating in Latin or Greek, by bringing them closer to a common phonetic notation.

## 3 Results

The results are shown in Table 4. The BLEU scores have been computed after replacing the let-

| ID | Epochs | Subwords | Train | Validation | Description |
|---|---|---|---|---|---|
| 1 | 45000 | 32768 external BPE | Med | News | early stop, when validation error started to increase |
| 2 | 28000 | 32768 external BPE | Med + News | Med | Train on NEWS as well for better fluency |
| 3 | 35000 | 32768 external BPE | Med + News | Med | 2 trained further |
| 4 | 28000 | 16384 T2T subwords | Med | News | repaired diacritics |
| 5 | 37000 | 16384 T2T subwords | Med | News | 4 trained further |
| 6 | 48000 | 32768 T2T subwords | Med | News | like 5, but with larger subwords dictionary |

Table 3: Transformer models trained. Models 1-3 used an external Byte Pair Encoding, whereas models 4-6 used the subwords in the tensor2tensor framework to achieve the capability of translating previously unseen words.

| ID | BLEU uncased | BLEU cased | Moses |
|---|---|---|---|
| 1 | 20.84 | 20.54 | 20.38 |
| 2 | 14.83 | 14.56 | 14.38 |
| 3 | 14.10 | 13.82 | 13.63 |
| 4 | **22.48** | **22.16** | **21.99** |
| 5 | 21.45 | 21.10 | 20.90 |
| 6 | 22.12 | 21.88 | 21.75 |
| **Ensemble** | **22.05** | **21.73** | **21.54** |

Table 4: BLEU scores evaluated using t2t-bleu from tensor2tensor and multi-bleu-detok from Moses

ters with cedilla-based diacritics both in the translation and in the reference translation with their correct comma-based version.

We have submitted two translations, the one produced by the model with ID=1 in Table 3 (cased BLEU=20.54) and the one produced by the entire ensemble (cased BLEU=21.73).

The run with ID=4 performed best with respect to the BLEU score. The output of the ensemble performed slightly worse than it (by about 0.5 BLEU points), but otherwise being almost equal to the second-best, ID=6.

## 4 Discussion and Conclusion

We chose to train on the MED corpora and test on NEWS based on the intuition that one can learn from medical texts how to generally translate arbitrary texts, up to the point where excessive specialization on the medical field is detrimental to the performance on the texts in other fields.

There are multiple ways to improve upon this work. The quality of the heuristic depends on the quality of the dictionary, so a straight-forward way would be to use a larger dictionary. The dictionary we have used had approx. 39000 word pairs, but only approx. 17000 Romanian words and approx. 20000 English words; there are multiple pairs for the same source word, when multiple translations exist. For comparison, the Explanatory Dictionary of the Romanian Language (DEX) contains 65000 word definitions.

Another way to improve would be replacing the manually engineered heuristic for evaluating the quality of the translations with one evaluation function learned with machine learning from sentence-aligned parallel corpora. The pair in the training set could then have the label 1 attached to it (with the meaning “correct translation”), whereas variations obtained by eliminating, inserting or changing in a random fashion words from the translation have the label 0 (“incorrect translation”) in the training set.

One reviewer suggested the models could have been combined in the decoder, by combining the word probabilities predictions – we did not try this yet. Each of the 6 members of the ensemble had its own decoder. The advantage in regarding the individual translators as atomic black boxes is that any type of translators can be used, including statistical and human translators. The obvious disadvantage is that in the ideal case the selected translation is the best among the translations to select from, but cannot outperform it; here, it selected reliably one of the best translations.

## References

UFAL medical corpus 1.0. https://ufal.mff.cuni.cz/ufal_medical_corpus. Accessed: 2018-07-24.

Cristian Grozea. 2012. Experiments and results with diacritics restoration in romanian. In *International Conference on Text, Speech and Dialogue*, pages 199–206. Springer.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*.

Long Zhou, Wenpeng Hu, Jiajun Zhang, and Chengqing Zong. 2017. Neural system combination for machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 378–384.

# LMU Munich's Neural Machine Translation Systems at WMT 2018

**Matthias Huck** and **Dario Stojanovski** and **Viktor Hangya** and **Alexander Fraser**
Center for Information and Language Processing
LMU Munich
Munich, Germany
{mhuck,stojanovski,hangyav,fraser}@cis.lmu.de

## Abstract

We present the LMU Munich machine translation systems for the English–German language pair. We have built neural machine translation systems for both translation directions (English→German and German→English) and for two different domains (the biomedical domain and the news domain). The systems were used for our participation in the WMT18 biomedical translation task and in the shared task on machine translation of news.[1,2]

The main focus of our recent system development efforts has been on achieving improvements in the biomedical domain over last year's strong biomedical translation engine for English→German (Huck et al., 2017a). Considerable progress has been made in the latter task, which we report on in this paper.

## 1 Introduction

Domain adaptation is one emphasis of the machine translation research conducted at the Center for Information and Language Processing at LMU Munich. Within the scope of our participation in the EU-funded *HimL* project (Haddow et al., 2017),[3] we were recently working on advancing the quality of machine translation for medical texts. The types of medical texts that we consider range from health information leaflets to professional biomedical research articles.

Some of our latest research towards medical domain adapation of neural translation systems is inspired by the "fine-tuning" approach in combination with high-quality in-domain data. Specifically, we conducted successive optimization runs to domain-adapt a neural translation model. The model was eventually deployed as the core component of the final English→German HimL translation engine in year 3 of the project (Y3).

In this paper, we give a brief technical overview of the HimL Y3 engine's neural translation model for English→German. We will show by how much the translation quality of medical texts improves compared to our previous year's WMT17 biomedical task submission (Huck et al., 2017a). We then proceed to compare with a Transformer model (Vaswani et al., 2017) that we have trained after the end of the HimL project. We find that the Transformer model performs even better than the HimL Y3 engine, which was based on Nematus (Sennrich et al., 2017) with a single hidden layer. The good result encouraged us to try out the Transformer in the other translation direction, German→English. We will also report the German→English results.

In addition to the English–German biomedical task, LMU Munich has participated in the WMT18 English–German news translation task (Bojar et al., 2018) in both translation directions. Our (supervised) news task systems are shortly described towards the end of the paper.[4]

## 2 Domain Adaptation

Medical texts differ in their style and in their topics from the typical content of many widely used training corpora, such as the parallel Europarl corpus (Koehn, 2005) or most of the large monolingual corpora that are distributed for the WMT shared task on machine translation of news (Bojar et al., 2018, 2017a, 2016, 2015). Medical documents also often contain a large amount of domain-specific technical terms in their vocabulary. Furthermore, sense shifts of words (away

[1] http://www.statmt.org/wmt18/biomedical-translation-task.html

[2] http://www.statmt.org/wmt18/translation-task.html

[3] http://www.himl.eu

[4] LMU's *unsupervised* machine translation system for the news task is described in a separate paper (Stojanovski et al., 2018).

Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64073

from their respective meaning in out-of-domain corpora) are common (Carpuat et al., 2013; Irvine et al., 2013).

Domain adaptation of conventional phrase-based machine translation systems is a well-explored research area. Several different effective solutions which may be used in order to domain-adapt a phrase-based system have been proposed in the literature. (Inter alia, cf. Huck et al. (2015) for a few interesting empirical results and a list of some major bibliographic references.) Machine translation in academic research labs and also in industry is however going through a paradigm shift away from phrase-based technology and on towards artificial neural network models. Neural machine translation (Sutskever et al., 2014; Bahdanau et al., 2014) is the new state of the art for basically all medium- to high-resource language pairs since around two to three years. The paradigm shift poses new challenges in domain adaptation, since most known techniques are rather specific to the phrase-based translation model and therefore cannot be readily applied to neural systems.

Domain adaptation of neural translation systems is a fresh and active field of scientific inquiry. The most wide-spread practical solution at present is referred to as “fine-tuning”. A baseline model is pre-trained by optimizing the neural model parameters on some large general corpus. Subsequently, training is simply continued on an in-domain corpus, usually with a smaller learning rate—i.e., in this second optimization run the parameters are initialized with the trained model parameters from the previous optimization. A crucial aspect is the availability of high-quality in-domain training data, or alternatively, the collection thereof. If a general-domain or out-of-domain neural model from a first optimization run already exists, then fine-tuning allows for quick adjustment of the model to a specific domain by means of a short continued optimization on an in-domain corpus, most often with less data than in the first run.

## 3 Neural Network Architectures

### 3.1 GRU Encoder-Decoder

We utilize the Nematus implementation (Sennrich et al., 2017) to build encoder-decoder NMT systems with attention and gated recurrent units (GRUs). Our architecture is flat, it has only one single hidden layer. We configure dimensions of 500 for the embeddings and 1024 for the hidden layer. We train with the Adam optimizer (Kingma and Ba, 2015), a learning rate of 0.0001, batch size of 50, and dropout with probability 0.2 applied to the hidden layer, but not to source, target, and embeddings. We validate every 10 000 updates and do early stopping when the validation cost has not decreased over ten consecutive control points.

### 3.2 Transformer

We use the Sockeye implementation of the Transformer (Hieber et al., 2017). For the German→English translation direction we train small Transformer models and for English→German big models as outlined in Vaswani et al. (2017). All models have six encoder and decoder layers. The size of the layers and the embeddings is 512 for the small models and 1024 for the big ones. The dimensionality of the feed-forward networks is 2048 (small) and 4096 (big). We use 8 attention heads for the small and 16 for the big models. The models are trained with the Adam optimizer with an initial learning rate of 0.0002. The learning rate is reduced by a factor of 0.7 if not improved for eight checkpoints. We checkpoint the models each 3 000 updates and do early stopping if perplexity has not improved for 32 checkpoints. We apply dropout of 0.1 as used by Vaswani et al. (2017). Additionally, we use label smoothing with a value of 0.1. We also tie the target and output embeddings. All models are trained with a word-level batch size of 4096.

## 4 Preprocessing

A linguistically informed, cascaded word segmentation technique is applied to the German side of the training data (Huck et al., 2017b). With a linguistically more sound word segmentation, we expect advantages over plain BPE segmentation in three important aspects: vocabulary reduction, reduction of data sparsity, and open vocabulary translation. The NMT system can learn linguistic word formation processes from the segmented data.

We cascade three different word splitting methods on the German side:

1. First we apply a suffix splitter that separates common German morphological suffixes from the word stems. Our suffix splitter is a modification of the German Snow-

ball stemming algorithm that separtates suffixes from the word stem, rather than stripping them.

2. Next we apply the empirical compound splitter as described by Koehn and Knight (2003).
3. We finally apply the Byte Pair Encoding (BPE) technique (Sennrich et al., 2016b) on top of the suffix-split and compound-split data in order to further reduce the vocabulary size.

Special marker symbols allow us to revert the segmentation in postprocessing when German is the target language.

Our linguistically informed word segmentation was already used on the target language side for LMU's participation in the WMT17 shared task on machine translation of news (Huck et al., 2017a). At WMT17, LMU's primary submission was ranked first in the human evaluation (Bojar et al., 2017a). We presume that the high human rating of LMU's WMT17 submission can mostly be attributed to our efforts toward better word segmentation. We anticipate similar benefits in the medical domain. Dedicated methods that tackle rich target-side morphology have also shown good results in phrase-based translation systems previously (Huck et al., 2017c). Future work on neural machine translation could for instance follow a two-step prediction paradigm (Conforti et al., 2018), or improve over our current version of linguistically informed word segmentation by means of a better linguistic analysis (Weissweiler and Fraser, 2017).

In the present work, the linguistically informed word segmentation is not only employed on the target side for English→German machine translation, but in German→English systems also on the source language side.

The English language side is always simply BPE-segmented.

We learn the compound split model and the BPE merge operations from Europarl and use this word segmentation and vocabulary for all corpora.

## 5 Systems: Medical Translation

### 5.1 English→German HimL Y3 System

The English→German HimL Y3 engine is based on a shallow GRU encoder-decoder model built with Nematus (Section 3.1). We apply an incremental training regime that is inspired by "fine-tuning" (Section 2). First, we train a model on parallel corpora from the WMT news task. We then successively refine the model and adapt it to the medical domain. Consecutive optimization runs are initialized with the respective previous model parameters. For each refinement step, we replace the training data, first with larger corpora, then with corpora that better match the domain.

The HimL tuning sets are used for validation, and we test separately on the Cochrane and NHS24 parts of the HimL devtest set.[5] The translation quality (in case-sensitive BLEU (Papineni et al., 2002)) of different system setups after several development stages is presented in the top section of Table 1. *WMT_parallel* denotes the Europarl, News Commentary, and Common Crawl parallel training data as provided for WMT17 by the organizers of the news translation shared task. *WMT_backtranslated_news_crawl* denotes Edinburgh's backtranslations of monolingual WMT News Crawl corpora from WMT16.[6] *Y3_base_general_data* is a large collection of English–German bitext used in the HimL project. *Cochrane-selected* and *NHS24-selected* denote synthetic data mixes from HimL whose content is automatically filtered to match the Cochrane or NHS24 use cases. Corpus statistics of the HimL training data and a more detailed description of the data selection procedure are provided by Bojar et al. (2017b) (Section 2.4 of HimL Deliverable D1.1).

We vary the learning rate during system development, as stated in the table. As a last step, we apply $n$-best list reranking ($n = 50$) with a right-to-left NMT model ("r2l reranking"). Ensembling did not yield any clear gains, so we deployed single models for English→German.

The bottom row of Table 1 contains the BLEU scores of our last year's primary system (Huck et al., 2017a) for the WMT17 biomedical task (Yepes et al., 2017). We improve over it by more than three points.

### 5.2 English→German Transformer System

We build Transformer models (Section 3.2) in order to evaluate whether they perform better than our Nematus-based HimL Y3 system.

For the English→German Transformer model, we train three separate models and ensemble them.

[5] http://www.himl.eu/test-sets

[6] http://data.statmt.org/rsennrich/wmt16_backtranslations/en-de/

| English→German | **Cochrane**<br>BLEU | **NHS24**<br>BLEU |
|---|---|---|
| WMT_parallel (*lrate = 0.0001*) | 31.5 | 28.9 |
| + WMT_parallel, WMT_backtranslated_news_crawl (*lrate = 0.0001*) | 29.8 | 27.6 |
| + UFAL_medical_shuffled_all (*lrate = 0.0001*) | 35.1 | 28.9 |
| + Y3_base_general_data (*lrate = 0.00001*) | 35.7 | 29.8 |
| + Cochrane-selected, NHS24-selected, 10 × UFAL_medical_indomain (*lrate = 0.00001*) | 38.6 | 33.0 |
| + r2l reranking (= HimL Y3) | **39.6** | **34.0** |
| Transformer single | 37.8 | 33.3 |
| Transformer ensemble | 39.0 | 34.1 |
| + r2l reranking | **40.3** | **35.5** |
| LMU WMT17 biomedical (Huck et al., 2017a) | **35.8** | **30.3** |

Table 1: English→German medical translation results on HimL devtest sets (case-sensitive BLEU). Extensions are applied incrementally. Particularly, in the top section of the table, which reports on HimL Y3 system engineering, we conduct successive model refinement by consecutively optimizing on different corpora. The middle section of the table reports on Transformer experiments. The row at the bottom provides the results of our WMT17 biomedical task system.

We also apply right-to-left reranking on these models as well. Because of time constraints we did not train a Transformer right-to-left model. Instead, we generated a 50-best list with the Transformer models and used the already trained Nematus right-to-left models for the reranking.

No incremental training regime or fine-tuning is applied to the Transformer system. We train on the same set of corpora that is also used in the last refinement step of the HimL Y3 system (Cochrane-selected, NHS24-selected, 10 × UFAL_medical_indomain).

The translation results with the English→German Transformer systems are presented in the middle section of Table 1. The Transformer outperforms our other systems.

We submitted three runs to the WMT18 biomedical translation shared task: the r2l-reranked Transformer (run1, primary); a Transformer ensemble without reranking (run2, contrastive); and the HimL Y3 system (run3, contrastive).

### 5.3 German→English Transformer System

Our German→English Transformer model is an ensemble of three separate models, like in the English→German translation direction. We use the same training corpus, but with source and target side switched. The preprocessing remains the same. Since German is the source language in this setup, our linguistically informed word segmentation technique is applied to the input side here.

The BLEU scores of the German→English Transformer without ensembling (single model) are 53.3 (Cochrane) and 41.7 (NHS24), respectively. The ensemble is reaching BLEU scores of 54.5 (Cochrane) and 42.2 (NHS24), which is a decent gain over the single model.

## 6 Systems: News Translation

### 6.1 English→German News Task System

For the shared task on machine translation of news, we did not build any updated system, but participated with our system from WMT17 (Bojar et al., 2017a). The system was trained under "constrained" conditions, employing only permissible resources as defined by the shared task organizers. Huck et al. (2017a) provide a detailed description, along with experimental results. In short, we conducted the following steps in an incremental training regime (with consecutive optimizations, in a similar manner as presented above for the HimL Y3 system):

1. Optimize a Europarl baseline model.
2. Add News Commentary and Common Crawl.
3. Add synthetic training data (Ueffing et al., 2007; Lambert et al., 2011; Huck et al., 2011; Huck and Ney, 2012; Sennrich et al., 2016a).
4. Fine-tune towards the domain of news articles. For that purpose, several `newstest` development sets are employed as a training corpus. The learning rate is decreased.
5. Rerank $n$-best list with a right-to-left neural model (Liu et al., 2016), which is trained for reverse word order (Freitag et al., 2013).

### 6.2 German→English News Task System

Finally, for the translation of news articles from German into English, we also trained a basic shallow GRU encoder-decoder system (cf. Section 3.1). The training data is a concatenation of Europarl, News Commentary, Common Crawl, and some synthetic data in the form of backtranslated English news texts. The German source side is preprocessed with our linguistically informed word segmentation (Section 4).

## 7 Conclusion

In this paper, we have described the steps we took to build a strong neural system for the translation of medical documents. Our English→German translation system was deployed within the HimL project. We used the system to participate in the WMT18 biomedical translation shared task. On HimL devtest sets, our WMT18 biomedical task systems outperforms our WMT17 submission system by more than three BLEU points.

Three aspects make our system effective in our view. (1.) We have high-quality in-domain training data at hand. (2.) A reliable preprocessing pipeline has been developed. (3.) A simple, but well-working domain adaptation method is known for neural machine translation.

The model architecture is also very important, as our additional Transformer experiments show: A less highly engineered Transformer model is on par with our deployed HimL project system.

Additionally to the English→German medical domain system, we have also briefly presented our system for the German→English translation direction and our WMT18 news task submissions.

## Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement № 644402 (HimL). This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement № 640550).

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints*, abs/1409.0473. Presented at ICLR 2015.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017a. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 Conference on Machine Translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Ondřej Bojar, Barry Haddow, David Mareček, Roman Sudarikov, Aleš Tamchyna, and Dušan Variš. 2017b. HimL Deliverable D1.1: Report on Building Translation Systems for Public Health Domain. Technical report. http://www.himl.eu/files/D1.1-report-on-building-translation-systems.pdf.

Marine Carpuat, Hal Daume III, Katharine Henry, Ann Irvine, Jagadeesh Jagarlamudi, and Rachel Rudinger. 2013. SenseSpotting: Never let your parallel data tie you to an old domain. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1435–1445, Sofia, Bulgaria. Association for Computational Linguistics.

Costanza Conforti, Matthias Huck, and Alexander Fraser. 2018. Neural Morphological Tagging of Lemma Sequences for Machine Translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (AMTA 2018), vol. 1: MT Research Track*, pages 39–53, Boston, MA, USA.

Markus Freitag, Minwei Feng, Matthias Huck, Stephan Peitz, and Hermann Ney. 2013. Reverse Word Order Models. In *Proceedings of the XIV Machine Translation Summit*, pages 159–166, Nice, France.

Barry Haddow, Alexandra Birch, Ondřej Bojar, Fabienne Braune, Colin Davenport, Alex Fraser, Matthias Huck, Michal Kašpar, Květoslava Kovaříková, Josef Plch, Anita Ramm, Juliane Ried, James Sheary, Aleš Tamchyna, Dušan Variš, Marion Weller, and Phil Williams. 2017. HimL: Health in my Language. In *Proceedings of the EAMT 2017 User Studies and Project/Product Descriptions*, page 33, Prague, Czech Republic.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Matthias Huck, Alexandra Birch, and Barry Haddow. 2015. Mixed-Domain vs. Multi-Domain Statistical Machine Translation. In *Proceedings of MT Summit XV, vol.1: MT Researchers' Track*, pages 240–255, Miami, FL, USA.

Matthias Huck, Fabienne Braune, and Alexander Fraser. 2017a. LMU Munich's Neural Machine Translation Systems for News Articles and Health Information Texts. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 315–322, Copenhagen, Denmark. Association for Computational Linguistics.

Matthias Huck and Hermann Ney. 2012. Pivot Lightly-Supervised Training for Statistical Machine Translation. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA 2012)*, San Diego, CA, USA.

Matthias Huck, Simon Riess, and Alexander Fraser. 2017b. Target-side Word Segmentation Strategies for Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Matthias Huck, Aleš Tamchyna, Ondřej Bojar, and Alexander Fraser. 2017c. Producing Unseen Morphological Variants in Statistical Machine Translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 369–375, Valencia, Spain. Association for Computational Linguistics.

Matthias Huck, David Vilar, Daniel Stein, and Hermann Ney. 2011. Lightly-Supervised Training for Hierarchical Phrase-Based Machine Translation. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 91–96, Edinburgh, Scotland. Association for Computational Linguistics.

Ann Irvine, John Morgan, Marine Carpuat, Hal Daumé III, and Dragos Munteanu. 2013. Measuring Machine Translation Errors in New Domains. *Transactions of the Association for Computational Linguistics*, 1:429–440.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the MT Summit X*, Phuket, Thailand.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 187–194, Budapest, Hungary. Association for Computational Linguistics.

Patrik Lambert, Holger Schwenk, Christophe Servan, and Sadaf Abdul-Rauf. 2011. Investigations on Translation Model Adaptation Using Monolingual Data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 284–293, Edinburgh, Scotland. Association for Computational Linguistics.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on Target-bidirectional Neural Machine Translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416, San Diego, CA, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, USA. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Dario Stojanovski, Viktor Hangya, Matthias Huck, and Alexander Fraser. 2018. The LMU Munich Unsupervised Machine Translation Systems. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Semi-supervised model adaptation for statistical machine translation. *Machine Translation*, 21(2):77–94.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Leonie Weissweiler and Alexander Fraser. 2017. Developing a Stemmer for German Based on a Comparative Analysis of Publicly Available Stemmers. In *Proceedings of the German Society for Computational Linguistics and Language Technology (GSCL)*, Berlin, Germany.

Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Karin Verspoor, Ondřej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, Pavel Pecina, Roland Roller, Amy Siu, Philippe Thomas, and Saskia Trescher. 2017. Findings of the WMT 2017 Biomedical Translation Shared Task. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

# Hunter NMT System for WMT18 Biomedical Translation Task: Transfer Learning in Neural Machine Translation

**Abdul Rafae Khan**[*] **Subhadarshi Panda**[*] **Jia Xu**, **Lampros Flokas**[†]
Hunter College, City University of New York
{ak4350,sp2951,jia.xu}@hunter.cuny.edu, lamflokas@cs.columbia.edu

## Abstract

This paper describes the submission of Hunter Neural Machine Translation (NMT) to the WMT'18 Biomedical translation task from English to French. The discrepancy between training and test data distribution brings a challenge to translate text in new domains. Beyond the previous work of combining in-domain with out-of-domain models, we found accuracy and efficiency gain in combining different in-domain models. We conduct extensive experiments on NMT with *transfer learning*. We train on different in-domain Biomedical datasets one after another. That means parameters of the previous training serve as the initialization of the next one. Together with a pre-trained out-of-domain News model, we enhanced translation quality with 3.73 BLEU points over the baseline. Furthermore, we applied ensemble learning on training models of intermediate epochs and achieved an improvement of 4.02 BLEU points over the baseline. Overall, our system is 11.29 BLEU points above the best system of last year on the EDP 2017 test set.

## 1 Introduction

Data-driven machine translation models assume the training data and test data have the same distribution and feature space (Koehn, 2009), which is rare in real-world applications (Olive et al., 2011). In statistical machine translation, a standard solution is to apply domain adaptation (Xu et al., 2007; Foster and Kuhn, 2007; Chu and Wang, 2018). For example, interpolating phrase or word probabilities in a sentence learned on in-domain and out-of-domain data and then computing their product. In NMT, we apply ensemble learning instead of interpolation. Moreover, we initialize neural networks with parameters trained with out-of-domain data. Studies show that this approach results in fast training and higher accuracy, such as in (Luong and Manning, 2015; Zoph et al., 2016; Freitag and Al-Onaizan, 2016).

| **Training** | **Bio'18** | **News'14** | **Bio'14** |
|---|---|---|---|
| $S_R$ $(M)$ | 2.8 | 41 | 19 |
| $S_P$ $(M)$ | 2.5 | 39 | 16 |
| $V_R$ $(B)$ | $61M/69M$ | 1.1/1.3 | 0.4/0.5 |
| V $(K)$ | 67/82 | 64/74 | 44/44 |

Table 1: **Raw and preprocessed data statistics for the three datasets used in the experiments.** $S_R$ is the sentences in the raw data, $S_P$ is the sentences in preprocessed data, $V_R$ is the running words and V is the vocabulary size. Running words & Vocabulary are for both source and target represented as *source/target*

These methods focus on combining an in-domain model with an out-of-domain model. Nonetheless, often, the training data is a mixture of multiple in-domain corpora and out-of-domain corpora. If one concatenates all the in-domain corpora to train a model, then training is more expensive for the memory and time. Furthermore, the distribution of one corpus may be closer than the others to the test set. Thus, the statistics of the closer corpus may vanish in the merged corpus.

The WMT'18 Biomedical translation task is a typical scenario. There are two sets of in-domain data: the Biomedical training set of WMT'18 (with 2.8M sentences) and WMT'14 (19 M), besides an out-of-domain training set on News (41 M), see Table 1. To separately train on WMT'18 and WMT'14 Biomedical data, a new challenge arises:

*How to efficiently combine the training on different in-domain training sets?*

To answer this question, this work presents an

[*] Both authors have contributed equally to this work.

[†] The author was working as a visiting student at Hunter College, CUNY

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 655–661
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64074

empirical study of efficient training on multiple in-domain and out-of-domain datasets. We applied *transfer learning* by training NMT systems with different datasets one after another carrying on the previous parameters. More precisely, we first initialize the NMT with the existing out-of-domain model trained on the out-of-domain News data. Then, we train the NMT with the in-domain Biomedical dataset of 2018. Afterward, we take the newly estimated parameters as the initialization and further train the NMT on the in-domain Biomedical dataset of 2014. In this way, a previous model's output initializes the parameters of the next model, so that we train on every single data set at a time instead all at once.

We further experimented with ensemble learning. We saved the model (checkpoint) after every epoch during training. Once training finishes, we performed *checkpoint ensembling* by picking various combinations of checkpoint outputs from the training on the last dataset.

We conduct our experiments on Biomedical translation task of WMT'18. We observe a significant accuracy improvement of 3.73 BLEU points for single models and 4.02 BLEU points for ensembles over our baseline trained with one in-domain dataset. While some of these improvements are due to differences in training data, preprocessing and hyper-parameters, most of the increase is due to the use of different data sets for initialization and subsequent training.

## 2 Related Work

In *domain adaptation* we aim at learning a model from a source data distribution which performs well on a different (but related) target data distribution. In machine translation domain adaptation arises when there is a large amount of out-of-domain data and a small amount of in-domain data. One technique to solve this issue is to increase the in-domain data size using different *data selection* methods (Moore and Lewis, 2010; Axelrod et al., 2011, 2012; Duh et al., 2013). They use in-domain language models to select in-domain data based on cross-entropy for SMT systems. (Xu et al., 2007; Foster and Kuhn, 2007) use a combination of feature weights and language model adaptation to build a domain-specific translation system. (Daumé III and Jagarlamudi, 2011) mine in-domain rare word translations using a comparable corpora in order to minimize the Out-of-Vocabulary (OOV) words. We aim to improve NMT accuracy and training efficiency by training on different corpora sequentially. Therefore, our method does not focus on selecting, mining, or interpolating in-domain data.

*Transfer learning* (Torrey and Shavlik, 2009; Pan and Yang, 2010) is the process where the model is trained by transferring the knowledge learned from an existing model. Domain adaptation also falls under this method. (Zoph et al., 2016) describe training a *parent model* in one language pair (out-of-domain data) which then can be used as an initialized *child model* for training another language pair (in-domain data). However in our work we train for the same language pair throughout the experiment. Another difference is that we apply transfer learning to train on two in-domain datasets one after the other.

(Luong and Manning, 2015) adapts an already existing NMT system to a new domain by further training on the in-domain data only. (Freitag and Al-Onaizan, 2016) in addition use checkpoint ensembling (Sennrich et al., 2016a; Koehn, 2017) to balance the performance on the in-domain data and out-of-domain data. In this paper, our goal is not to adapt from out-of-domain to in-domain data. We aim to empirically investigate training on multiple in-domain datasets to improve in-domain performance, which has not been discussed in above previous work. We show that during time-sensitive system development, training on in-domain datasets one after another has its pragmatical use. It significantly improves the translation accuracy over the training on a single dataset, i.e. 3.73 BLEU points, and is also more efficient than training on all in-domain datasets at once.

## 3 Background

NMT is an approach to machine translation using a neural network which takes as an input a source sentence $(x_1, .., x_t, .., x_I)$ and generates its translation $(y_1, .., y_{t'}, .., y_{I'})$, where $x_t$ and $y_{t'}$ are source and target words respectively. The dominant approach to NMT till recent times encodes the input sequence and subsequently generates a variable length translated sequence using recurrent neural networks (RNN) (Bahdanau et al., 2014; Sutskever et al., 2014; Cho et al., 2014).

We use the sequence to sequence learning architecture by (Gehring et al., 2017), which uses convolutional neural networks (CNNs) instead of RNNs. This model has three components, namely, encoder, decoder and an attention mechanism.

The encoder combines a short sequence of neighboring words into a single representation. Convolutions are carried out consecutively in multiple layers to get the final representation of each word. For each input word to the encoder, the state at each convolutional layer is informed by the corresponding state in the previous layer and its neighbors determined by a fixed window. Even with a few layers, the final representation of a word generated by the encoder may only be informed by partial sentence context.

There are significant computational advantages to this paradigm. All words at one depth can be processed in parallel, even combined into one massive tensor operation that can be efficiently parallelized on a GPU.

The decoder of the CNN based NMT model calculates the decoder state conditioning on the sequence of the $k$ most recent previous words. The states of the decoder are computed in a sequence of convolutional layers and depend only on the input context, with no dependence on the previous decoder state. The attention mechanism in CNN based architecture is essentially unchanged from the RNN based model.

## 4 Transfer Learning

A domain $\mathcal{D}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$ where $X \in \mathcal{X}$ is a training sample. If two domains are different then they must have different feature spaces or different marginal probabilities. *Transfer learning* is defined as follows:

**Definition 1.** *Given a source domain $\mathcal{D}_S$ and a learning task $\mathcal{T}_S$,* transfer learning *aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.*

In the above definition, a domain is a pair $\mathcal{D} = \{\mathcal{X}, P(X)\}$. Thus the condition $\mathcal{D}_S \neq \mathcal{D}_T$ implies that either $\mathcal{X}_S \neq \mathcal{X}_T$ or $P_S(X) \neq P_T(X)$. One category of transfer learning is *transductive transfer learning* where the source and the target tasks are the same but the domain is different. This can be further categorized into two cases. For the machine translation scenario, these are that either the feature spaces between domains are different, $\mathcal{X}_S \neq \mathcal{X}_T$ (e.g., News and Biomedical), or their marginal distributions are different, $P(X_S) \neq P(X_T)$ (e.g., Biomedical'14 and Biomedical'18).

We apply transfer learning to learn the target predictive function $f_T(\cdot)$ in both the above cases. We use the CNN based architecture described in Section 3 to train the NMT model parameters. First we train for the case when the the domain feature spaces are different, i.e., $\mathcal{X}_S \neq \mathcal{X}_T$. We consider $\mathcal{X}_S$ as News data and $\mathcal{X}_T$ as Biomedical'18 data, since they represent two different domains.

For this training, we re-use a pre-trained system (Gehring et al., 2017) on News corpus and continue training on Biomedical'18 corpus. We re-use a pre-trained system because the training on News corpus requires a large training time[1]. For training the CNN based NMT, we only use the vocabulary of Biomedical'18 for simplicity.

We then apply transfer learning for the second case: when the marginal distributions of $\mathcal{X}_S$ and $\mathcal{X}_T$ are different ($P(X_S) \neq P(X_T)$). Now we can consider $\mathcal{X}_S$ as Biomedical'18 data and $\mathcal{X}_T$ as Biomedical'14 data. This is because they are in the same domain with different marginal distributions. We continue training the model learned on the Biomedical'18 data further with Biomedical'14 data. Again, we just use the vocabulary of the latter data for training.

The use of transfer learning significantly increases the translation quality (see Figure 1). The BLEU score obtained using transfer learning from News to Biomedical'18 data is shown in the middle part of the plot (Bio'18). The BLEU curve reaches a peak of 30.97% in BLEU score in this part of transfer learning.

Furthermore, upon using Biomedical'14 data, we get additional improvement, as shown in the right side of the plot (Bio'14). We get the highest peak of 34.83% in BLEU score. The learned parameters from one set of data are transferred while training on another set and enhance the translation quality.

During training, we evaluate the performance of the model after every epoch using a development set from the Biomedical domain. Our system is prone to over-fitting as the Biomedical (2014 and 2018) training data sets that we use are significantly smaller (see Table 1) as compared to News. Generally over-fitting means that the model performs excellent on the training data, but worse on

[1] 37 days using 8 GPUs on WMT' 18 EN-FR (Gehring et al., 2017)

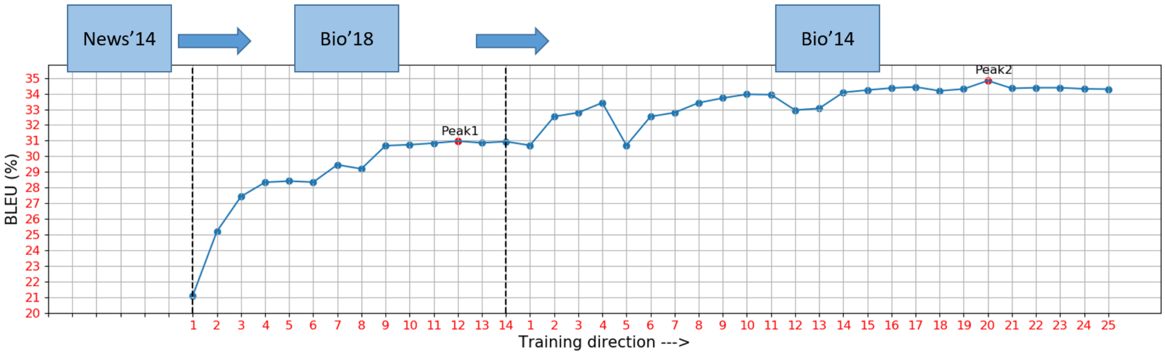


Figure 1: **BLEU[%] during transfer learning** The results are calculated on EDP'17 test data. The $x$-axis shows the epoch number during training.

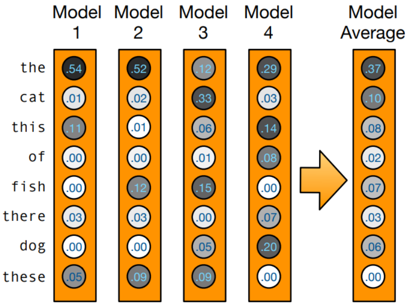


Figure 2: Combining predictions from an ensemble of models (Koehn, 2017)

any other unseen data. To overcome this problem, we use ensemble learning.

More concretely, we save the models (checkpoints) after every epoch of training. We use the predictions of multiple checkpoints instead of just one checkpoint. We perform this ensemble of models for different epochs, called checkpoint ensembling, as follows: Each model defined by a checkpoint generates a probability distribution over target vocabulary. We average these distributions to obtain a combined probability distribution. Then we use the combined distribution to predict the output word. See Figure 2 for an illustration. Checkpoint ensembling is computationally less expensive than multi-run ensembling, another typical approach for ensembling NMT models. In multi-run ensembling, each system is built in a completely different training run. In checkpoint ensembling, we get all the checkpoints from a single run.

# 5 Experiments

This section describes the datasets, tools, and settings used for the Biomedical translation task.

| Data Set | Dev Data | | Test Data |
|---|---|---|---|
| | *Kh* | *Kh*+HIML | EDP'17 |
| $S_R$ | 500 | 2011 | 500 |
| $S_P$ | 500 | 2011 | 499 |
| $V_R$ $(K)$ | 11/13 | 37/46 | 13/15 |
| V $(K)$ | 3/3 | 5/6 | 3/3 |
| OOV | 154/177 | 329/499 | 271/366 |

Table 2: **Development and test data stats.** *Kh* refers to the Khresmoi development data. $S_R$ is the sentences in the raw data, $S_P$ is the sentences in preprocessed data, $V_R$ is the running words, V is the vocabulary size and OOV is the unique Out-Of-Vocabulary words. Running words, Vocabulary & Out-Of-Vocabulary words are represented as *source/target*.

## 5.1 Datasets

We used the WMT'18 Biomedical shared task English-French (EN-FR) data for training. In this paper, this data is the UFAL medical corpus[2]. We also used WMT'14 Biomedical EN-FR data (PatTR[3] only) as additional in-domain data. For out-of-domain training, we used WMT'14 News EN-FR training data. We validated each training epoch on Khresmoi and HIML development datasets. We use the WMT'17 EDP (Yepes et al., 2017) as test data to evaluate. Statistics for the development and test data sets is mentioned in Table 1 and Table 2.

## 5.2 Preprocessing

We tokenized and true-cased the training, development and test data using the script provided by Moses.[4] We only used sentences no longer than 80 words (for training data only). Then we learned byte pair encoding (BPE) by combining

[2] https://ufal.mff.cuni.cz/ufal_medical_corpus
[3] http://www.cl.uni-heidelberg.de/statnlpgroup/pattr/
[4] https://github.com/moses-smt/mosesdecoder

the WMT'18 Biomedical EN and FR training corpus. We used a script from (Sennrich et al., 2016b) with $89,500$ merge operations. This gave a dictionary size of $63.6K$ for EN and $74.1K$ for FR. We also applied BPE to WMT'14 Biomedical data resulting in dictionary sizes $67K$ and $81.9K$ for EN and FR respectively. Our best model uses the latter dictionaries for translation.

### 5.3 Training Details

To train our systems we used the open source toolkit Fairseq[5] which provides an implementation of the CNN based NMT model (Gehring et al., 2017). We trained three different sets of models: (1) training on WMT'18 Biomedical data only, (2) training on the WMT'14 News, followed by training on WMT'18 Biomedical data, and (3) training on the WMT'14 News, then training on WMT'18 Biomedical and then training on WMT'14 Biomedical. Apart from this we also trained using different development sets which include Khresmoi and Khresmoi+HIML.

For the training of all systems, we used a learning rate of $0.25$ and dropout of $0.2$. We fixed the maximum batch size to be $4000$ tokens. On a Tesla V100 with 16 GB RAM, it took about $40$ hours for training on WMT'18 Biomedical till convergence and $500$ hours for training on WMT'14 Biomedical for $25$ epochs.

Another possible experiment can be to combine the two in-domain datasets and then train. This experiment however takes 22 days for training of 25 epochs as compared to 1.7 days for completely training on WMT'18 Biomedical data. Therefore we trained on the WMT'18 Biomedical data till convergence and subsequently trained on the larger WMT'14 Biomedical data for some epochs. Additionally we also saved on the training time by using a pre-trained model on WMT'14 News data to initialize the system parameters. Details of training time (per epoch) for each dataset are mentioned in Table 3. Combining datasets is also memory intensive as compared to training on separate data.

### 5.4 Decoding Details

For translation, we used either the best epoch (which gave the minimum loss on the development data) or an ensemble of different epochs during the training process. The Fairseq tool provides a simple method to use specific epoch(s) for translation. We removed BPE before evaluation. We tuned the decoding beam size and used a beam size of 12 for all translations. The best model settings were then used to translate the WMT'18 test datasets (EDP & Medline).

| **Data Set** | **Training Time**<br>per Epoch (hrs) |
|---|---|
| News'14 | 41 |
| Bio'18 | 2.5 |
| Bio'14 | 19 |
| Bio'18 + Bio'14 | 21.5 |

Table 3: **Training time for each dataset.** Training time is for a single epoch in hours.

## 6 Results

Table 4 shows BLEU scores for different experiments with and without ensemble. The arrow shows the flow of training the translation model, for example, "news14 $\rightarrow$ bio18 $\rightarrow$ bio14" means the system was first trained on WMT'14 News data, then on WMT'18 Biomedical data and finally on WMT'14 Biomedical data. The single model results are obtained using the best checkpoints (the best checkpoint is the one which gave minimum loss on the development data) for each experiment and the ensemble results are obtained using the best ensemble of multiple checkpoints. We evaluate the translations using the `multi-bleu.pl` script from Moses.

For the baseline method (Exp 1) we trained only using WMT'18 Biomedical data. The single best model gave 31.10% in BLEU score. Ensemble of different checkpoints did not improve the results, therefore it has the same BLEU score as single model. In Exp 2 we used a pre-trained model on the WMT'14 News and continued training on WMT'18 Biomedical data. The single model gave the BLEU score of $30.97\%$ which is less than Exp 1, but ensembling improved the BLEU score to $31.18\%$. On further training on another in-domain WMT'14 Biomedical data (Exp 3), the single best model greatly improves the performance with a BLEU score of $34.83\%$. Ensemble of different checkpoints improves this further to $35.12\%$. This is an improvement of $3.73$ BLEU points for the single model and $4.02$ BLEU points from the baseline experiment (Exp 1). The best model uses checkpoints 2, 4 and 24.

The best system for WMT'17 (Exp $a$) on EN-

[5] https://github.com/pytorch/fairseq

| No. | Experiment | BLEU [%] | |
|---|---|---|---|
| | | Single | Ensemble |
| $a$ | WMT'17 best system | 27.04* | – |
| 1 | bio18 (*baseline*) | 31.10 | 31.10 |
| 2 | news14 → bio18 | 30.97 | 31.18 |
| 3 | news14 → bio18 → bio14 | 34.83 | **35.12** , **38.33*** |

Table 4: **BLEU scores for different models on EDP'17 test data.** *Single* is the single model which gave minimum loss on the Khresmoi development set. Results with (*) are calculated using multi-eval tool. All other results are calculated using multi-bleu tool.

| No. | Experiment | Dev Set | BLEU [%] |
|---|---|---|---|
| 1 | news14 → bio18 | Khresmoi | 30.97 |
| 2 | news14 → bio18 | Khresmoi +HIML | 29.23 |

Table 5: **Results of different development sets for tuning all the models.** BLEU scores are calculated on EDP'17 test data.

| Checkpoint Number | | | | | | | | | | | | | | BLEU[%] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | |
| ● | | | | | | | | | | | ● | | | 30.38 |
| ● | | | | ● | | | | | ● | | ● | | | 30.93 |
| | | | | ● | | | | | ● | | ● | | | **31.18** |
| | | | | | | | | | | ● | ● | ● | ● | 30.98 |
| | | | | | | | | | ● | ● | ● | ● | ● | 30.86 |
| | | | | | | | | ● | ● | ● | ● | ● | ● | 30.38 |
| | | | | | | | ● | ● | ● | ● | ● | ● | ● | 30.90 |
| | | | | | | ● | ● | ● | ● | ● | ● | ● | ● | 31.05 |

Table 6: **BLEU scores for different checkpoint ensembles for Exp 2 (Table 4).** Cells with dots in each row show checkpoints for ensemble. Checkpoint 12 gave the minimum loss on the development data.

FR EDP test data gave 27.04% in BLEU score using `mteval-v13a.pl` script from Moses. Using the same script our best model (Exp 3 in Table 4) gave 38.33% in BLEU score. This is an improvement of **11.29** BLEU points.

We also tested with using different development sets for tuning the model. The results are in Table. 5. We get better results when using Khresmoi development data as compared to a combined Khresmoi and HIML development data.

Apart from this we also carried out ensemble experiments to compare which checkpoint combination gives the best result. Only checkpoints for Exp 2 in Table 4 are considered. Among the 14 checkpoints output during training process, checkpoint 12 gave the minimum loss on the development data. We tried a several checkpoint combinations of these 14 checkpoints, some of these are mentioned in Table 6. The best checkpoint combination is 5, 10 and 12.

## 7 Conclusion

We studied training on different in-domain datasets and found significant improvement by consecutively training on an out-of-domain dataset (WMT'14 News) and multiple in-domain datasets (WMT'18 Biomedical and then WMT'14 Biomedical). We successfully applied transfer learning by initializing parameters of NMT with a previous model. Together with ensemble learning, we achieved 4.02 BLEU points enhancement over our baseline. Overall, our system is 11.29 BLEU points better than the best WMT'17 system.

## 8 Acknowledgement

This research was partially supported by National Science Foundation (NSF) Award No. 1747728 and National Science Foundation of China (NSFC) Award No. 61672524. We express our gratitude to the provost office, the deans office, the department of computer science, and the research foundation of Hunter College at CUNY for partially funding our research. We also thank the computer science department of the Graduate Center CUNY to provide facilities to our team.

## References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Stroudsburg, PA, USA. Association for Computational Linguistics.

Amittai Axelrod, QingJun Li, and William D. Lewis. 2012. Applications of data selection via cross-entropy difference for real-world statistical machine translation. In *IWSLT*, pages 201–208. ISCA.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111. Association for Computational Linguistics.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258*.

Hal Daumé III and Jagadeesh Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 407–412. Association for Computational Linguistics.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 678–683.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *Computing Research Repository*, abs/1612.06897.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *Computing Research Repository*, abs/1705.03122.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn. 2017. Neural machine translation. *Computing Research Repository*, abs/1709.07809.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224. Association for Computational Linguistics.

Joseph Olive, Caitlin Christianson, and John McCary. 2011. *Handbook of natural language processing and machine translation: DARPA global autonomous language exploitation*. Springer Science & Business Media.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Lisa Torrey and Jude Shavlik. 2009. Transfer learning.

Jia Xu, Yonggang Deng, Yuqing Gao, and Hermann Ney. 2007. Domain dependent statistical machine translation. In *MT Summit*.

Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Karin Verspoor, Ondrej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, et al. 2017. Findings of the wmt 2017 biomedical translation shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation.

# UFRGS Participation on the WMT Biomedical Translation Shared Task

**Felipe Soares**
Instituto de Informtica - UFRGS
Porto Alegre - RS - Brazil
felipe.soares@inf.ufrgs.br

**Karin Becker**
Instituto de Informtica - UFRGS
Porto Alegre - RS - Brazil
karin.becker@inf.ufrgs.br

## Abstract

This paper describes the machine translation systems developed by the Universidade Federal do Rio Grande do Sul (UFRGS) team for the biomedical translation shared task. Our systems are based on statistical machine translation and neural machine translation, using the Moses and OpenNMT toolkits, respectively. We participated in four translation directions for the English/Spanish and English/Portuguese language pairs. To create our training data, we concatenated several parallel corpora, both from in-domain and out-of-domain sources, as well as terminological resources from UMLS. Our systems achieved the best BLEU scores according to the official shared task evaluation.

## 1 Introduction

In this paper, we present the system developed at the Universidade Federal do Rio Grande do Sul (UFRGS) for the Biomedical Translation shared task in the Third Conference on Machine Translation (WMT18), which consists in translating scientific texts from the biological and health domain. In this edition of the shared task, six language pairs are considered: English/Chinese, English/French, English/German, English/Portuguese, English/Romanian, and English/Spanish.

Our participation in this task considered the English/Portuguese and English/Spanish language pairs, with translations in both directions. For that matter, we developed two machine translation (MT) systems: one based on statistical machine translation (SMT), using Moses (Koehn et al., 2007), and one using neural machine translation (NMT), using OpenNMT (Klein et al., 2017).

This paper is structured as follows: Section 3 details the language resources used to train our translation models. Section 4 contains the description of the experimental settings of our SMT and NMT models, including the pre-processing step performed to comply with the shared task guidelines. In Section 5 we present the results and briefly discuss the main findings. Section 6 contains the conclusions and directions of future works to improve our models.

## 2 Related Works

Most of related works in biomedical machine translation used SMT models to perform automatic translation. Aires et al. (2016) developed a phrase-based SMT that differs significantly from the usual Moses toolkit, especially by not analyzing phrases at word level and adopting a translation score that is a tuned weighted average between the translation model and the language model, instead of the traditional log-linear approach.

Costa-Jussà et al. (2016) employed Moses SMT to perform automatic translation integrated with a neural character-based recurrent neural network for model re-ranking and bilingual word embeddings for out of vocabulary (OOV) resolution. Given the 1000-best list of SMT translations, the RNN performs a rescoring and selects the translation with the highest score. The OOV resolution module infers the word in the target language based on the bilingual word embedding trained on large monolingual corpora. Their reported results show that both approaches can improve BLEU scores, with the best results given by the combination of OOV resolution and RNN re-ranking. Similarly, Ive et al. (2016) also used the n-best output from Moses as input to a re-ranking model, which is based on a neural network that can handle vocabularies of arbitrary size.

In the last WMT biomedical translation chal-

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 662–666
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64075

lenge (2017) (Yepes et al., 2017), the submission that achieved the best BLEU scores for the FR/EN language pair on the EDP dataset, in both directions, was based on NMT models developed in the University of Kyoto (Cromieres, 2016). For the other datasets, the submission from the University of Edinburgh (Sennrich et al., 2017) achieved the best BLEU scores with their NMT models based on the Nematus implementation with BPE tokenization and the use of parallel and backtranslated data.

## 3 Resources

In this section, we describe the language resources used to train both models, which are from two main types: corpora and terminological resources.

### 3.1 Corpora

We used both in-domain and general domain corpora to train our systems. For general domain data, we used the books corpus (Tiedemann, 2012), which is available for several languages, included the ones we explored in our systems, and the JRC-Acquis (Tiedemann, 2012). As for in-domain data, we included several different corpora:

- The corpus of full-text scientific articles from Scielo (Soares et al., 2018a), which includes articles from several scientific domains in the desired language pairs, but predominantly from biomedical and health areas.

- A subset of the UFAL medical corpus[1], containing the Medical Web Crawl data for the English/Spanish language pair.

- The EMEA corpus (Tiedemann, 2012), consisting of documents from the European Medicines Agency.

- A corpus of theses and dissertations abstracts (BDTD) (Soares et al., 2018b) from CAPES, a Brazilian governmental agency responsible for overseeing post-graduate courses. This corpus contains data only for the English/Portuguese language pair.

- A corpus from Virtual Health Library[2] (BVS), containing also parallel sentences for the language pairs explored in our systems.

Table 1 depicts the original number of parallel segments according to each corpora source. In Section 3.1, we detail the pre-processing steps performed on the data to comply with the task evaluation.

| Corpus | Sentences | |
|---|---|---|
| | EN/ES | EN/PT |
| Books | 93,471 | - |
| UFAL | 286,779 | - |
| Full-text Scielo | 425,631 | 2.86M |
| JRC-Acquis | 805,757 | 1.64M |
| EMEA | - | 1.08M |
| CAPES-BDTD | - | 950,252 |
| BVS | 737,818 | 631,946 |
| Total | 2.37M | 7.19M |

Table 1: Original size of individual corpora used in our experiments

### 3.2 Terminological Resources

Regarding terminological resources, we extracted parallel terminologies from the Unified Medical Language System[3] (UMLS). For that matter, we used the MetamorphoSys application provided by U.S. National Library of Medicine (NLM) to subset the language resources for our desired language pairs. Our approach is similar to what was proposed by Perez-de Viñaspre and Labaka (2016).

Once the resource was available, we imported the MRCONSO RRF file to an SQL database to split the data in a parallel format in the two language pairs. Table 2 shows the number of parallel concepts for each pair.

| Language Pair | Concepts |
|---|---|
| EN/ES | 14,399 |
| EN/PT | 26,194 |

Table 2: Number of concepts from UMLS for each language pair

## 4 Experimental Settings

In this section, we detail the pre-processing steps employed as well as the architecture of the SMT and NMT systems.

[1] https://ufal.mff.cuni.cz/ufal_medical_corpus

[2] http://bvsalud.org/

[3] https://www.nlm.nih.gov/research/umls/

### 4.1 Pre-processing

As detailed in the description of the biomedical translation task, the evaluation is based on texts extracted from Medline. Since one of our corpora, the one comprised of full-text articles from Scielo, may contain a considerable overlap with Medline data, we decided to employ a filtering step in order to avoid including such data.

The first step in our filter was to download metadata from Pubmed articles in Spanish and Portuguese. For that matter, we used the Ebot utility[4] provided by NLM using the queries *POR[la]* and *ESP[la]*, retrieving all results available. Once downloaded, we imported them to an SQL database which already contained the corpora metadata. To perform the filtering, we used the *pii* field from Pubmed to match the Scielo unique identifiers or the title of the papers, which would match documents not from Scielo.

Once the documents were matched, we removed them from our database and partitioned the data in training and validation sets. Table 3 contains the final number of sentences for each language pair and partition.

| Language | Train | Dev |
|---|---|---|
| EN/ES | 2.35M | 22,670 |
| EN/PT | 7.17M | 24,206 |

Table 3: Final corpora size for each language pair

### 4.2 SMT System

We used the popular Moses toolkit (Koehn et al., 2007) to train our SMT system for the two language pairs. As training parameters, we followed the Moses baseline steps[5] to train four MT systems (i.e. one for each translation direction).

Regarding training, we used the Amazon AWS spot virtual machines with 24 cores and 60GB of RAM, and used parallelization as much as possible to reduce training time and the associated cost.

### 4.3 NMT System

As for the NMT system, we employed the OpenNMT toolkit (Klein et al., 2017) to train four MT systems, one for each translation direction. Tokenization was performed by the supplied OpenNMT algorithm. Regarding network parametrization, the following settings were used, while all other parameters were set as default:

- Encoder type: bidirectional recurrent neural network
- Decoder type: Seq2Seq with attention (default)
- Word vector size: 600
- Layers (encoder and decoder): 4
- RNN size: 800
- Batch size: 64
- Vocabulary size: 50000

To train our system, we used the Azure virtual machines with a single NVIDIA Tesla V100 GPU. The models with the best perplexity value were chosen as final models. During translation, OOV words were replace by their original word in the source language, all other OpenNMT options for translation were kept as default.

## 5 Experimental Results

We now detail the results achieved by our SMT and NMT systems on the official test data used in the shared task. Table 4 shows the BLEU scores (Papineni et al., 2002) for both systems and for the submissions made by other teams.

Our submissions achieved the best results for all translation directions we participated, with remarkable BLEU scores for the ES/EN and PT/EN pairs. When compared to the other teams, our results presented similar behavior, with higher scores when English was the target language, which may be explained by the poor English morphosyntactic system. For the English/Spanish pair, the SMT system presented slightly better results than the NMT one, probably due to the dictionary size used in the NMT.

Regarding the superior results achieved, we expect that the large parallel corpora used in our experiments played an essential role. Although we did not use the provided Scielo abstracts corpus (Neves et al., 2016), we used a newer parallel corpus also from Scielo, but comprised of full-text articles (Soares et al., 2018a), which overlaps with the abstracts, but contains more data.

In addition to the biomedical and health corpora, we employed two out-of-domain corpora that we assumed to have a similar structure to scientific texts: the books and the JRC-Acquis (Tiedemann, 2012). We decided not to use the

[4] https://www.ncbi.nlm.nih.gov/Class/PowerTools/eutils/ebot/ebot.cgi

[5] http://www.statmt.org/moses/?n=moses.baseline

| Team, Runs | EN/ES | EN/PT | ES/EN | PT/EN |
|---|---|---|---|---|
| UFRGS run1 (NMT) | 39.62 | **39.43** | 43.31 | **42.58** |
| UFRGS run2 (SMT) | **39.77** | **39.43** | **43.41** | **42.58** |
| TGF TALP UPC run1 | - | - | 40.49 | 39.49 |
| TGF TALP UPC run2 | - | - | 39.06 | 38.54 |
| UHH-DS run1 | 31.32 | 34.92 | 36.16 | 41.84 |
| UHH-DS run2 | 31.05 | 34.19 | 35.17 | 41.80 |
| UHH-DS run3 | 31.33 | 34.49 | 36.05 | 41.79 |

Table 4: Official BLEU scores for the English/Spanish and English/Portuguese language pairs in both translation directions. Bold numbers indicate the best result for each direction.

large Europarl corpus (Koehn, 2005), since it is comprised of speeches transcripts, which do not follow the usual structure of scientific texts.

## 6 Conclusions

We presented the UFRGS machine translation systems for the biomedical translation shared task in WMT18. For our submissions, we trained SMT and NMT systems for all four translation directions for the English/Spanish and English/Portuguese language pairs.

For model building, we included several corpora from biomedical and health domain, and from out-of-domain data that we considered to have similar textual structure, such as JRC-Acquis and books. Prior training, we also pre-processed our corpora to ensure, or at least minimize the risk, of including Medline data in our training set, which could produce biased models, since the evaluation was carried out on texts extracted from Medline.

Our systems achieved the best results in this shared task for the translation directions we participated, which we attribute to the high quality corpora used and their size.

Regarding future work, we are planning on optimizing our systems by studying the following methods:

- BPE tokenization: as stated by Sennrich et al. (2016b), the use of byte pair encoding tokenization can help to tackle the issue of OOV words by using subword units. We expect that this approach can provide better results for our NMT system on biomedical data, since this domain contains terminologies that are usually based on the use of affixes.

- Backtranslation: the use of synthetic data from back-translation of monolingual proved to be able to increase NMT performance (Sennrich et al., 2016a) by providing additional training data.

- Multilingual training: a study from Google (Johnson et al., 2017) showed that using multilingual data when training NMT systems can improve translation performance, especially when using a many-to-one scheme (i.e. several source languages and one target language). We expect that systems trained using (ES+PT)→EN, for instance, may produce better results due to the similarity between Portuguese and Spanish.

## References

José Aires, Gabriel Lopes, and Luís Gomes. 2016. English-portuguese biomedical translation task using a genuine phrase-based statistical machine translation approach. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 456–462.

Marta R Costa-Jussà, Cristina España-Bonet, Pranava Madhyastha, Carlos Escolano, and José AR Fonollosa. 2016. The talp–upc spanish–english wmt biomedical task: Bilingual embeddings and char-based neural language model rescoring in a phrase-based system. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 463–468.

Fabien Cromieres. 2016. Kyoto-nmt: a neural machine translation implementation in chainer. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 307–311.

Julia Ive, Aurélien Max, and François Yvon. 2016. Limsi's contribution to the wmt'16 biomedical translation task. In *First Conference on Machine Translation*, volume 2, pages 469–476.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5(1):339–351.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Mariana Neves, Antonio Jimeno Yepes, and Aurlie Nvol. 2016. The scielo corpus: a parallel corpus of scientific publications for biomedicine. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. *arXiv preprint arXiv:1708.00726*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Felipe Soares, Viviane Moreira, and Karin Becker. 2018a. A Large Parallel Corpus of Full-Text Scientific Articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Felipe Soares, Gabrielli Yamashita, and Michel Anzanello. 2018b. A parallel corpus of theses and dissertations abstracts. In *The 13th International Conference on the Computational Processing of Portuguese (PROPOR 2018)*, Canela, Brazil. Springer International Publishing.

Jrg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Olatz Perez-de Viñaspre and Gorka Labaka. 2016. Ixa biomedical translation system at wmt16 biomedical translation task. In *Proceedings of the First Conference on Machine Translation*, pages 477–482, Berlin, Germany. Association for Computational Linguistics.

Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Karin Verspoor, Ondrej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, et al. 2017. Findings of the wmt 2017 biomedical translation shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247.

# Neural Machine Translation with the Transformer and Multi-Source Romance Languages for the Biomedical WMT 2018 task

**Brian Tubay and Marta R. Costa-jussà**
TALP Research Center, Universitat Politcnica de Catalunya, Barcelona
brian.alcides.tubay.alvarez@alu-etsetb.upc.edu,marta.ruiz@upc.edu

## Abstract

The Transformer architecture has become the state-of-the-art in Machine Translation. This model, which relies on attention-based mechanisms, has outperformed previous neural machine translation architectures in several tasks. In this system description paper, we report details of training neural machine translation with multi-source Romance languages with the Transformer model and in the evaluation frame of the biomedical WMT 2018 task. Using multi-source languages from the same family allows improvements of over 6 BLEU points.

## 1 Introduction

Neural Machine Translation (NMT) (Bahdanau et al., 2015) proved to be competitive with the encoder-decoder architecture based on recurrent neural networks and attention. After this architecture, new proposals based on convolutional neural networks (Gehring et al., 2017) or only attention-based mechanisms (Vaswani et al., 2017) appeared. The latter architecture has achieved great success in Machine Translation (MT) and it has already been extended to other tasks such as Parsing (Kaiser et al., 2017), Speech Recognition [1], Speech Translation (Cros et al., 2018), Chatbots (Costa-jussà et al., 2018) among others.

However, training with low resources is still a big drawback for neural architectures and NMT is not an exception (Koehn and Knowles, 2017). To face low resource scenarios, several techniques have been proposed, like using multi-source (Zoph and Knight, 2016), multiple languages (Johnson et al., 2017) or unsupervised techniques (Lample et al., 2018; Artetxe et al., 2018), among many others.

In this paper, we use the Transformer enhanced with the multi-source technique to participate in the Biomedical WMT 2018 task, which can be somehow considered a low-resourced task, given the large quantity of data that it is required for NMT. Our multi-source enhancement is done only with Romance languages. The fact of using similar languages in a multi-source system may be a factor towards improving the final system which ends up with over 6 BLEU points of improvement over the single source system.

## 2 The Transformer architecture

The Transformer model is the first NMT model relying entirely on self-attention to compute representations of its input and output without using recurrent neural networks (RNN) or convolutional neural networks (CNN).

RNNs read one word at a time, having to perform multiple steps before generating an output that depends on words that are far away. But it has been demonstrated that the more steps required, the harder it is to the network to learn how to make these decisions (Bahdanau et al., 2015). In addition, given the sequential nature of the RNNs, it is difficult to fully take advantage of modern computing devices such as Tensor Processing Units (TPUs) or Graphics Processing Units (GPUs) which rely on parallel processing. The Transformer is an encoder-decoder model that was conceived to solve these problems.

The encoder is composed of three stages. In the first stage input words are projected into an embedded vector space. In order to capture the notion of token position within the sequence, a positional encoding is added to the embedded input vectors. Without positional encodings, the output of the multi-head attention network would be the same for the sentences “I love you more than her”

[1] https://tensorflow.github.io/tensor2tensor/tutorials\/asr$_$with$_$transformer.html

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 667–670
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64076

and "I love her more than you". The second stage is a multi-head self-attention. Instead of computing a single attention, this stage computes multiple attention blocks over the source, concatenates them and projects them linearly back onto a space with the initial dimensionality. The individual attention blocks compute the scaled dot-product attention with different linear projections. Finally a position-wise fully connected feed-forward network is used, which consists of two linear transformations with a ReLU activation (Vinod Nair, 2010) in between.

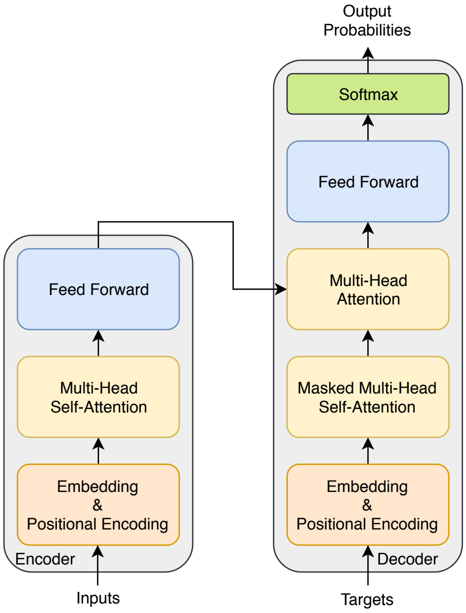


Figure 1: Simplified diagram of the Transformer model

The decoder operates similarly, but generates one word at a time, from left to right. It is composed of five stages. The first two are similar to the encoder: embedding and positional encoding and a masked multi-head self-attention, which unlike in the encoder, forces to attend only to past words. The third stage is a multi-head attention that not only attends to these past words, but also to the final representations generated by the encoder. The fourth stage is another position-wise feed-forward network. Finally, a softmax layer allows to map target word scores into target word probabilities. For more specific details about the architecture, refer to the original paper (Vaswani et al., 2017).

## 3 Multi-Source translation

Multi-source translation consists in exploiting multiple text inputs to improve NMT (Zoph and Knight, 2016). In our case, we are using this approach in the Transformer architecture described above and using only inputs from the same language family.

## 4 Experiments

In this section we report details on the database, training parameters and results.

### 4.1 Databases and Preprocessing

The experimental framework is the Biomedical Translation Task (WMT18)[2]. The corpus used to train the model are the one provided for the task for the selected languages pairs: Spanish-to-English (es2en), French-to-English (fr2en) and Portuguese-to-English (pt2en). Sources are mainly from Scielo and Medline and detailed in Table 3.

| Training | Scielo | Medline | Total |
|---|---|---|---|
| es2en | 713127 | 285358 | 998485 |
| fr2en | 9127 | 612645 | 621772 |
| pt2en | 634438 | 74267 | 708705 |
| all2en | 1356692 | 972270 | 2328962 |

Table 3: Corpus Statistics (number of segments).

Validation sets were taken from *Khresmoi development data*[3], as recommended in the task description. Each validation dataset contains 500 sentence pairs. Test sets were the ones provides by the task for the previous year competition (WMT17[4]).

Preprocessing relied on three basic steps: tokenization, truecasing and limiting sentence length to 80 words. Words were segmented by means of Byte-Pair Encoding (BPE) (Sennrich et al., 2015).

### 4.2 Parameters

The system was implemented using OpenNMT in PyTorch (Klein et al., 2017) with the hyperparameters suggested in the website [5]. Other parameters used in training are defined in Table 4. Both single-language systems and multi-source system

[2] http://www.statmt.org/wmt18/biomedical-translation-task.html

[3] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2122

[4] http://www.statmt.org/wmt17/biomedical-translation-task.html

[5] http://opennmt.net/OpenNMT-py/FAQ.html

| **System** | **es2en** | | **pt2en** | | **fr2en** | |
|---|---|---|---|---|---|---|
| | WMT17 | WMT18 | WMT17 | WMT18 | WMT17 | WMT18 |
| Best performing system | 37.49 | **43.31** | 43.88 | **42.58** | - | **25.78** |
| Single-Language | 39.35 | 39.06 | 44.31 | 38.54 | 31.75 | 19.42 |
| Multi-Language | **40.11** | 40.49 | **45.55** | 39.49 | **38.31** | **25.78** |

Table 1: Trained systems results for WMT17 and WMT18 official test sets.

| | |
|---|---|
| **Spanish** | Utilizando la base de datos Epistemonikos, la cual es mantenida mediante bsquedas realizadas en 30 bases de datos, identificamos seis revisiones sistemticas que en conjunto incluyen 36 estudios aleatorizados pertinentes a la pregunta. |
| **Single-Language** | Using the Epistemonikos database, which is maintained through searches in 30 databases, we identified six systematic reviews including 36 randomized studies relevant to the question. |
| **Multi-Language** | Using the Epistemonikos database, which is maintained through searches in 30 databases, we identified six systematic reviews that altogether include 36 randomized studies relevant to the question. |
| **Portuguese** | Os resultados dos modelos de regresso mostraram associao entre os fatores de correo estimados e os indicadores de adequao propostos |
| **Single-Language** | Regression models showed an association between estimated correction factors and the proposed adequacy indicators. |
| **Multi-Language** | The results of the regression models showed an association between the estimated correction factors and the proposed adequacy indicators. |
| **French** | (Traduit par Docteur Serge Messier). |
| **Single-Language** | [Doctor Serge Messier]. |
| **Multi-Language** | [(Translated by Doctor Serge Messier)]. |

Table 2: Spanish/Portuguese/French to English examples for WMT18

were trained with same architecture and parameters.

| **Hparam** | **Text-to-Text** |
|---|---|
| Encoder layers | 6 |
| Decoder layers | 6 |
| Batch size | 4096 |
| Adam optimizer | $\beta_1 = 0.9 \quad \beta_2 = 0.998$ |
| Attention heads | 8 |

Table 4: Training parameters.

We trained three single-language systems, one for each language pair. We required 14 epochs for the Spanish-to-English system (7 hours of training), 16 epochs for the French-to-English system (9 hours of training), and 17 epochs for the Portuguese-to-English system (7 hours of training). For the multi-source system, which concatenated the three parallel corpus together, we required 11 epochs (23 hours of training). We stopped training when the validation accuracy did not increase in two consecutive epochs.

### 4.3 Results

Best ranking systems from WMT17 and WMT18 are shown in Table 1, except for French-to-English WMT17 since the references for this set are not available. For this pair, we used 1000 sentences from the Khresmoi development data. Table 1 shows BLEU results for the baseline systems, the single-language and multi-source approaches.

The Transformer architecture outperforms WMT17 best system. Results become even better with the system is trained with the common corpus of Romance languages, what we call the multi-source approach. The latter is consistent with the universal truth that more data equals better results, even if the source language is not the same.

Finally, Table 2 shows some examples of the output translations.

## 5 Conclusions

The main conclusions of our experiments are that the multi-source inputs of the same family applied to the Transformer architecture can improve the single input. Best improvements achieve an increase of 6 BLEU points in translation quality.

## Acknowledgments

Authors would like to thank Noe Casas for his valuable comments. This work is supported in part by the Spanish Ministerio de Economía y Competitividad, the European Regional Development Fund and the Agencia Estatal de Investigación, through the postdoctoral senior grant Ramón y Cajal, the contract TEC2015-69266-P (MINECO/FEDER,EU) and the contract PCIN-2017-079 (AEI/MINECO).

## References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Marta R. Costa-jussà, Álvaro Nuez, and Carlos Segura. 2018. Experimental research on encoder-decoder architectures with attention for chatbots. *Computación y Sistemas*.

Laura Cros, Carlos Escolano, José A. R. Fonollosa, and Marta R. Costa-jussà. 2018. End-to-end speech translation with the transformer. *Submitted to IberSpeech*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017. One model to learn them all. *arXiv preprint arXiv:1706.05137*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proc. of the 1st Workshop on Neural Machine Translation*, pages 28–39, Vancouver.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Geoffrey E. Hinton Vinod Nair. 2010. Rectified linear units improve restricted boltzmann machines. In *27th International Conference on Machine Learning*.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *NAACL-HLT 2016*, pages 30–34.

# Results of the WMT18 Metrics Shared Task

**Qingsong Ma**
Tencent-MIG
AI Evaluation & Test Lab
qingsong.mqs@gmail.com

**Ondřej Bojar**
Charles University
MFF ÚFAL
bojar@ufal.mff.cuni.cz

**Yvette Graham**
Dublin City University
ADAPT
graham.yvette@gmail.com

## Abstract

This paper presents the results of the WMT18 Metrics Shared Task. We asked participants of this task to score the outputs of the MT systems involved in the WMT18 News Translation Task with automatic metrics. We collected scores of 10 metrics and 8 research groups. In addition to that, we computed scores of 8 standard metrics (BLEU, SentBLEU, chrF, NIST, WER, PER, TER and CDER) as baselines. The collected scores were evaluated in terms of system-level correlation (how well each metric's scores correlate with WMT18 official manual ranking of systems) and in terms of segment-level correlation (how often a metric agrees with humans in judging the quality of a particular sentence relative to alternate outputs). This year, we employ a single kind of manual evaluation: direct assessment (DA).

## 1 Introduction

Accurate machine translation (MT) evaluation is important for measuring improvements in system performance. Human evaluation can be costly and time consuming, and it is not always available for the language pair of interest. Automatic metrics can be employed as a substitute for human evaluation in such cases, metrics that aim to measure improvements to systems quickly and at no cost to developers. In the usual set-up, an automatic metric carries out a comparison of MT system output translations and human-produced reference translations to produce a single overall score for the system.[1] Since there exists a large number of possible approaches to producing quality scores for translations, it is sensible to carry out a meta-evaluation of metrics with the aim to estimate their accuracy as a substitute for human assessment of translation quality. The Metrics Shared Task[2] of WMT annually evaluates the performance of automatic machine translation metrics in their ability to provide a substitute for human assessment of translation quality.

Again, we keep the two main types of metric evaluation unchanged from the previous years. In *system-level* evaluation, each metric provides a quality score for the whole translated test set (usually a set of documents, in fact). In *segment-level* evaluation, a score is assigned by a given metric to every individual sentence.

The underlying texts and MT systems come from the News Translation Task (Bojar et al., 2018, denoted as Findings 2018 in the following). The texts were drawn from the news domain and involve translations to/from Chinese (zh), Czech (cs), German (de), Estonian (et), Finnish (fi), Russian (ru), and Turkish (tr), each paired with English, making a total of 14 language pairs.

A single form of golden truth of translation quality judgement is used this year:

- In *Direct Assessment* (DA) (Graham et al., 2016), humans assess the quality of a given MT output translation by comparison with a reference translation (as opposed to the source and reference). DA is the new standard used in WMT

[1] The availability of a reference translation is the key difference between our task and *MT quality estimation*, where no reference is assumed.

[2] `http://www.statmt.org/wmt18/metrics-task.html`, starting with Koehn and Monz (2006) up to Bojar et al. (2017)

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 671–688
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64077

News Translation Task evaluation, requiring only monolingual evaluators.

As in last year's evaluation, the official method of manual evaluation of MT outputs is no longer "relative ranking" (RR, evaluating up to five system outputs on an annotation screen relative to each other) as this was changed in 2017 to DA. For system-level evaluation, we thus use the Pearson correlation $r$ of automatic metrics with DA scores. For segment-level evaluation, we re-interpret DA judgements as relative comparisons and use Kendall's $\tau$ as a substitute, see below for details and references.

Section 2 describes our datasets, i.e. the sets of underlying sentences, system outputs, human judgements of translation quality and also participating metrics. Sections 3.1 and 3.2 then provide the results of system and segment-level metric evaluation, respectively. We discuss the results in Section 4.

## 2 Data

This year, we provided the task participants with one test set along with reference translations and outputs of MT systems. Participants were free to choose which language pairs they wanted to participate and whether they reported system-level, segment-level scores or both.

### 2.1 Test Sets

We use the following test set, i.e. a set of source sentences and reference translations:

**newstest2018** is the test set used in WMT18 News Translation Task (see Findings 2018), with approximately 3,000 sentences for each translation direction (except Chinese and Estonian which have 3,981 and 2,000 sentences, resp.). newstest2018 includes a single reference translation for each direction.

### 2.2 Translation Systems

The results of the Metrics Task are likely affected by the actual set of MT systems participating in a given translation direction. For instance, if all of the systems perform similarly, it will be more difficult, even for humans, to distinguish between the quality of translations. If the task includes a wide range of systems of varying quality, however, or systems are quite different in nature, this could in some way make the task easier for metrics, with metrics that are more sensitive to certain aspects of MT output performing better.

This year, the MT systems included in the Metrics Task were:

**News Task Systems** are machine translation systems participating in the WMT18 News Translation Task (see Findings 2018).[3]

**Hybrid Systems** are created automatically with the aim of providing a larger set of systems against which to evaluate metrics, as in Graham and Liu (2016). Hybrid systems were created for newstest2018 by randomly selecting a pair of MT systems from all systems taking part in that language pair and producing a single output document by randomly selecting sentences from either of the two systems. In short, we create 10K hybrid MT systems for each language pair.

Excluding the hybrid systems, we ended up with 149 systems across 14 language pairs.

### 2.3 Manual MT Quality Judgments

Direct Assessment (DA) was employed as the "golden truth" to evaluate metrics again this year. The details of this method of human evaluation is provided in two sections for system-level evaluation (Section 2.3.1) and segment-level evaluation (Section 2.3.2).

The DA manual judgements were provided by MT researchers taking part in WMT tasks, a number of in-house human evaluators at Amazon and crowd-sourced workers on Amazon Mechanical Turk.[4] Only judgements from workers who passed DA's quality control mechanism were included in the final datasets used to compute system and segment-level scores employed as a gold standard in the Metrics Task.

[3] One system for tr-en was unfortunately omitted from the first run of human evaluation in the News Translation Task and due to time constraints was subsequently omitted from the Metrics Task evaluation, Alibaba-Ensemble.

[4] https://www.mturk.com

### 2.3.1 System-level Manual Quality Judgments

In the system-level evaluation, the goal is to assess the quality of translation of an MT system for the whole test set. Our manual scoring method, DA, nevertheless proceeds sentence by sentence, aggregating the final score as described below.

**Direct Assessment (DA)** This year the translation task employed monolingual direct assessment (DA) of translation adequacy (Graham et al., 2013; Graham et al., 2014a; Graham et al., 2016). Since sufficient levels of agreement in human assessment of translation quality are difficult to achieve, the DA setup simplifies the task of translation assessment (conventionally a bilingual task) into a simpler monolingual assessment. In addition, DA avoids bias that has been problematic in previous evaluations introduced by assessment of several alternate translations on a single screen, where scores for translations had been unfairly penalized if often compared to high quality translations (Bojar et al., 2011). DA therefore employs assessment of individual translations in isolation from other outputs.

Translation adequacy is structured as a monolingual assessment of similarity of meaning where the target language reference translation and the MT output are displayed to the human assessor. Assessors rate a given translation by how adequately it expresses the meaning of the reference translation on an analogue scale corresponding to an underlying 0-100 rating scale.[5]

Large numbers of DA human assessments of translations for all 14 language pairs included in the News Translation Task were collected from researchers and from workers on Amazon's Mechanical Turk, via sets of 100-translation hits to ensure sufficient repeat assessments per worker, before application of strict quality control measures to filter out assessments from poor performers.

In order to iron out differences in scoring strategies attributed to distinct human assessors, human assessment scores for translations were standardized according to an individual judge's overall mean and standard deviation score. Final scores for MT systems were computed by firstly taking the average of scores for individual translations in the test set (since some were assessed more than once), before combining all scores for translations attributed to a given MT system into its overall adequacy score. The gold standard for system-level DA evaluation is thus what is denoted "Ave $z$" in Findings 2018 (Bojar et al., 2018).

Finally, although it was necessary to apply a sentence length restriction in WMT human evaluation prior to the introduction of DA, the simplified DA setup does not require restriction of the evaluation in this respect and no sentence length restriction was applied in DA WMT18.

### 2.3.2 Segment-level Manual Quality Judgments

Segment-level metrics have been evaluated against DA annotations for the newstest2018 test set. This year, a standard segment-level DA evaluation of metrics, where each translation is assessed a minimum of 15 times, was unfortunately not possible due to insufficient number of judgements collected. DA judgements were therefore converted to relative ranking judgements (daRR) to produce results. This is the same strategy as carried out for some out-of-English language pairs in last year's evaluation.

**daRR** When we have at least two DA scores for translations of the same source input, it is possible to convert those DA scores into a relative ranking judgement, if the difference in DA scores allows conclusion that one translation is better than the other. In the following, we will denote these re-interpreted DA judgements as "daRR", to distinguish it clearly from the "RR" golden truth used in the past years.

Since the analogue rating scale employed by DA is marked at the 0-25-50-75-100 points, the difference in DA scores we employ to distinguish translations that are better/worse than one another is 25 points. Note that we rely on judgements collected from known-reliable volunteers and crowd-sourced workers who passed DA's quality control mechanism. Any inconsistency that could arise from re-

[5] The only numbering displayed on the rating scale are extreme points 0 and 100%, and three ticks indicate the levels of 25, 50 and 75 %.

| | DA>1 | Ave | DA pairs | daRR |
|---|---|---|---|---|
| cs-en | 2,491 | 3.6 | 13,223 | 5,110 |
| de-en | 2,995 | 11.4 | 192,702 | 77,811 |
| en-cs | 1,586 | 4.9 | 15,311 | 5,413 |
| en-de | 2,150 | 5.3 | 47,041 | 19,711 |
| en-et | 1,035 | 13.6 | 90,755 | 32,202 |
| en-fi | 1,481 | 5.3 | 30,613 | 9,809 |
| en-ru | 2,954 | 6.2 | 54,260 | 22,181 |
| en-tr | 707 | 3.4 | 4,750 | 1,358 |
| en-zh | 3,915 | 6.5 | 86,286 | 28,602 |
| et-en | 2,000 | 11.2 | 118,066 | 56,721 |
| fi-en | 2,972 | 5.4 | 39,127 | 15,648 |
| ru-en | 2,916 | 4.9 | 31,361 | 10,404 |
| tr-en | 2,991 | 4.5 | 24,325 | 8,525 |
| zh-en | 3,952 | 7.2 | 97,474 | 33,357 |

Table 1: Number of judgements for DA converted to daRR data; "DA>1" is the number of source input sentences in the manual evaluation where at least two translations of that same source input segment received a DA judgement; "Ave" is the average number of translations with at least one DA judgement available for the same source input sentence; "DA pairs" is the number of all possible pairs of translations of the same source input resulting from "DA>1"; and "daRR" is the number of DA pairs with an absolute difference in DA scores greater than the 25 percentage point margin.

liance on DA judgements collected from low quality crowd-sourcing, for example, is thus prevented.

From the complete set of human assessments collected for the News Translation Task, all possible pairs of DA judgements attributed to distinct translations of the same source were converted into daRR better/worse judgements. Distinct translations of the same source input whose DA scores fell within 25 percentage points (which could have been deemed equal quality) were omitted from the evaluation of segment-level metrics. Conversion of scores in this way produced a large set of daRR judgements for all language pairs, shown in Table 1 due to combinatorial advantage of extracting daRR judgements from all possible pairs of translations of the same source input.

**Kendall's Tau-like Formulation for daRR** We measure the quality of metrics' segment-level scores against the daRR golden truth using a Kendall's Tau-like formulation, which is an adaptation of the conventional Kendall's Tau coefficient. Since we do not have a total order ranking of all translations we use to evaluate metrics, it is not possible to apply conventional Kendall's Tau given the current daRR human evaluation setup (Graham et al., 2015). Vazquez-Alvarez and Huckvale (2002) also note that a genuine pairwise comparison is likely to lead to more stable results for segment-level metric evaluation.

Our Kendall's Tau-like formulation, $\tau$, is as follows:

$$\tau = \frac{|Concordant| - |Discordant|}{|Concordant| + |Discordant|} \qquad (1)$$

where *Concordant* is the set of all human comparisons for which a given metric suggests the same order and *Discordant* is the set of all human comparisons for which a given metric disagrees. The formula is not specific with respect to ties, i.e. cases where the annotation says that the two outputs are equally good.

The way in which ties (both in human and metric judgement) were incorporated in computing Kendall $\tau$ has changed across the years of WMT Metrics Tasks. Here we adopt the version used in the last years' WMT17 daRR evaluation (but not earlier). For a detailed discussion on other options, see also Macháček and Bojar (2014).

Whether or not a given comparison of a pair of distinct translations of the same source input, $s_1$ and $s_2$, is counted as a concordant (Conc) or disconcordant (Disc) pair is defined by the following matrix:

| | | Metric | | |
|---|---|---|---|---|
| | | $s_1 < s_2$ | $s_1 = s_2$ | $s_1 > s_2$ |
| Human | $s_1 < s_2$ | Conc | Disc | Disc |
| | $s_1 = s_2$ | – | – | – |
| | $s_1 > s_2$ | Disc | Disc | Conc |

In the notation of Macháček and Bojar (2014), this corresponds to the setup used in WMT12 (with a different underlying method of manual judgements, RR):

|  |  | Metric |  |  |
|---|---|---|---|---|
|  | WMT12 | < | = | > |
| Human | < | 1 | -1 | -1 |
|  | = | X | X | X |
|  | > | -1 | -1 | 1 |

The key differences between the evaluation used in WMT14–WMT16 and evaluation used in WMT17 and WMT18 are (1) the move from RR to daRR and (2) the treatment of ties.[6] In the years 2014-2016, ties in metrics scores were not penalized. With the move to daRR, where the quality of the two candidate translations is deemed substantially different and no ties in human judgements arise, it makes sense to penalize ties in metrics' predictions in order to promote discerning metrics.

Note that the penalization of ties makes our evaluation asymmetric, dependent on whether the metric predicted the tie for a pair where humans predicted < or >. It is now important to interpret the meaning of the comparison identically for humans and metrics. For error metrics, we thus reverse the sign of the metric score prior to the comparison with human scores: higher scores have to indicate better translation quality. In WMT18, we did this for ITER and the original authors did this for CharacTER.

To summarize, the WMT18 Metrics Task for segment-level evaluation:

- excludes all human ties (this is already implied by the construction of DARR from DA judgements),
- counts metric's ties as a *Discordant* pairs,
- ensures that error metrics are first converted to the same orientation as the human judgements, i.e. higher score indicating higher translation quality.

We employ bootstrap resampling (Koehn, 2004; Graham et al., 2014b) to estimate confidence intervals for our Kendall's Tau formulation, and metrics with non-overlapping 95% confidence intervals are identified as having statistically significant difference in performance.

## 2.4 Participants of the Metrics Shared Task

Table 2 lists the participants of the WMT18 Shared Metrics Task, along with their metrics. We have collected 10 metrics from a total of 8 research groups.

The following subsections provide a brief summary of all the metrics that participated. The list is concluded by our baseline metrics in Section 2.4.9.

As in last year's task, we asked participants whose metrics are publicly available to provide links to where the code can be accessed. Table 3 provides links for metrics that participated in WMT18 that are publicly available for download.

### 2.4.1 BEER

BEER (Stanojević and Sima'an, 2015) is a trained evaluation metric with a linear model that combines features sub-word feature indicators (character n-grams) and global word order features (skip bigrams) to get language agnostic and fast to compute evaluation metric. BEER has participated in previous years of the evaluation task.

### 2.4.2 Blend

BLEND incorporates existing metrics to form an effective combined metric, employing SVM regression for training and DA scores as the gold standard. For to-English language pairs, incorporated metrics include 25 lexical based metrics and 4 other metrics. Since some lexical based metrics are simply different variants of the same metric, there are only 9 kinds of lexical based metrics, namely BLEU, NIST, GTM, METEOR, ROUGE, Ol, WER, TER and PER. 4 other metrics are CharacTER, BEER, DPMF and ENTF.

BLEND has participated in the Metrics Task in WMT17. This year, BLEND follows its setup in WMT17, but enlarges the training data since there are some data available in WMT17. For to-English language pairs, there are 9280 sentences as training data, while1620 sentences for English-Russian (en-ru). Experiments show the performance of BLEND can be improved if the training data increases.

BLEND is flexible to be applied to any language pairs if incorporated metrics support the

[6] Due to an error in the write-up for WMT17 (errata to follow), this second change was not properly reflected in the paper, only in the evaluation scripts.

| Metric | Seg-level | Sys-level | Hybrids | Participant |
|---|---|---|---|---|
| BEER | • | ⊘ | ⊘ | ILLC – University of Amsterdam (Stanojević and Sima'an, 2015) |
| BLEND | • | ⊘ | ⊘ | Tencent-MIG-AI Evaluation & Test Lab (Ma et al., 2017) |
| CharacTer | • | • | • | RWTH Aachen University (Wang et al., 2016a) |
| ITER | • | • | ⋆ | Jadavpur University (Panja and Naskar, 2018) |
| meteor++ | • | ⊘ | ⊘ | Peking University (Guo et al., 2018) |
| RUSE | • | ⊘ | ⊘ | Tokyo Metropolitan University (Shimanaka et al., 2018) |
| UHH_TSKM | • | ⊘ | ⊘ | (Duma and Menzel, 2017) |
| YiSi-* | • | ⊘ | ⊘ | NRC (Lo, 2018) |

Table 2: Participants of WMT18 Metrics Shared Task. "•" denotes that the metric took part in (some of the language pairs) of the segment- and/or system-level evaluation and whether hybrid systems were also scored. "⊘" indicates that the system-level and hybrids are implied, simply taking arithmetic average of segment-level scores. "⋆" indicates that the original ITER system-level scores should be calculated as the *micro-average* of segment-level scores but we calculate them as simple macro-averaged for the hybrid systems. See the ITER paper for more details.

| | |
|---|---|
| BEER | `https://github.com/stanojevic/beer` |
| BLEND | `https://github.com/qingsongma/blend` |
| CharacTer | `https://github.com/rwth-i6/CharacTER` |
| RUSE | `https://github.com/Shi-ma/RUSE` |
| YiSi-0, incl. -1 and -1_srl | `http://chikiu-jackie-lo.org/home/index.php/yisi` |
| Baselines: | `http://github.com/moses-smt/mosesdecoder` |
| BLEU, NIST | `scripts/generic/mteval-v13a.pl` |
| CDER, PER, TER, WER | `mert/evaluator` ("Moses scorer") |
| sentBLEU | `mert/sentence-bleu` |
| chrF, chrF+ | `https://github.com/m-popovic/chrF` |

Table 3: Metrics available for public download that participated in WMT18. Most of the baseline metrics are available with Moses, relative paths are listed.

specific language pair and DA scores are available.

### 2.4.3 CharacTer

CharacTer (Wang et al., 2016b; Wang et al., 2016a), identical to the 2016 setup, is a character-level metric inspired by the commonly applied translation edit rate (TER). It is defined as the minimum number of character edits required to adjust a hypothesis, until it completely matches the reference, normalized by the length of the hypothesis sentence. CharacTer calculates the character-level edit distance while performing the shift edit on word level. Unlike the strict matching criterion in TER, a hypothesis word is considered to match a reference word and could be shifted, if the edit distance between them is below a threshold value. The Levenshtein distance between the reference and the shifted hypothesis sequence is computed on the character level. In addition, the lengths of hypothesis sequences instead of reference sequences are used for normalizing the edit distance, which effectively counters the issue that shorter translations normally achieve lower TER.

Similarly to other character-level metrics, CharacTer is applied to non-tokenized outputs and references, which also holds for this year's submission.

This year tokenization was carried out for en-ru hypotheses and reference before calculating the scores, since this results in large improvements in terms of correlations. For other language pairs a tokenizer was not used for pre-processing. A python library was used for calculating the Levenshtein distance, so that the metric is now about 7 times faster than before.

#### 2.4.4 ITER

ITER (Panja and Naskar, 2018) is an improved Translation Edit/Error Rate (TER) metric. In addition to the basic edit operations in TER (insertion, deletion, substitution and shift), ITER also allows stem matching and uses optimizable edit costs and better normalization.

Note that for segment-level evaluation, we reverse the sign of the score, so that better translations get higher scores. For system-level confidence, we calculate the system-level scores for hybrids systems slightly differently than the original ITER definition would require. We use the unweighted arithmetic average of segment-level scores (macro-average) whereas ITER would use the micro-average.

#### 2.4.5 meteor++

METEOR++ (Guo et al., 2018) is metric based on Meteor (Denkowski and Lavie, 2014), adding explicing treatment of "copy-words", i.e. words that are likely to be preserved across all paraphrases of a sentence in a given language.

#### 2.4.6 RUSE

RUSE (Shimanaka et al., 2018) is a perceptron regressor based on three types of sentence embeddings: Infersent, Quick-Thought and Universal Sentence Encoder, designed with the aim to utilize global sentence information that cannot be captured by local features based on character or word n-grams. The sentence embeddings come from pre-trained models and the regression itself is trained on past manual judgements in WMT shared tasks.

#### 2.4.7 UHH_TSKM

UHH_TSKM (Duma and Menzel, 2017) is a non-trained metric utilizing kernel functions, i.e. methods for efficient calculation of overlap of substructures between the candidate and the reference translations. The metric uses both sequence kernels, applied on the tokenized input data, together with tree kernels, that exploit the syntactic structure of the sentences. Optionally, the match can also be performed for the candidate and a pseudo-reference (i.e. a translation by another MT system) or for the source sentence and the candidate back-translated into the source language.

#### 2.4.8 YiSi-0, YiSi-1 and YiSi-1_srl

The YISI metrics (Lo, 2018) are recently proposed semantic MT evaluation metrics inspired by MEANT_2.0 (?). Specifically, YISI-1 is identical to MEANT_2.0-NOSRL which featured in the WMT17 Metrics Task.

YISI-1 also successfully served in the parallel corpus filtering task. Some details are provided in the system description paper (?).

YISI-1 measures the relative lexical semantic similarity (weighted word embeddings cosine similarity aggregated into $n$-grams similarity) of the candidate and reference translations, optionally taking the shallow semantic structure ("srl") into account. YISI-0 is a degenerate resource-free version using the longest common character substring, instead of word embeddings cosine similarity, to measure the word similarity of the candidate and reference translations.

#### 2.4.9 Baseline Metrics

As mentioned by Bojar et al. (2016), Metrics Task occasionally suffers from "loss of knowledge" when successful metrics participate only in one year.

We attempt to avoid this by regularly evaluating also a range of "baseline metrics":

- **Mteval.** The metrics BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) were computed using the script `mteval-v13a.pl`[7] that is used in the OpenMT Evaluation Campaign and includes its own tokenization. We run `mteval` with the flag `--international-tokenization` since it performs slightly better (Macháček and Bojar, 2013).

- **Moses Scorer.** The metrics TER (Snover et al., 2006), WER, PER and CDER (Leusch et al., 2006) were produced by the Moses scorer, which is used in Moses model optimization. To tokenize the sentences, we used the standard tokenizer script as available in Moses toolkit. When tokenizing, we also convert all outputs to lowercase.

[7] `http://www.itl.nist.gov/iad/mig/tools/`

Since Moses scorer is versioned on Github, we strongly encourage authors of high-performing metrics to add them to Moses scorer, as this will ensure that their metric can be easily included in future tasks.

- **SentBLEU.** The metric sentBLEU is computed using the script sentence-bleu, a part of the Moses toolkit. It is a smoothed version of BLEU that correlates better with human judgements for segment-level. Standard Moses tokenizer is used for tokenization.

- **chrF** The metrics chrF and chrF+ (Popović, 2015; Popović, 2017) are computed using their original Python implementation.

  We run `chrF++.py` with the parameters `-nw 0 -b 3` to obtain the **chrF** score and with `-nw 0 -b 1` to obtain the chrF+ score. Note that chrF intentionally removes all spaces before matching the $n$-grams, detokenizing the segments but also concatenating words.

  We originally planned to use the chrF implementation which was recently made available in Moses Scorer but it mishandles Unicode characters for now.

The baselines serve in system and segment-level evaluations as customary: BLEU, TER, WER, PER and CDER for system-level only; sentBLEU for segment-level only and chrF for both.

**Chinese word segmentation** is unfortunately not supported by the tokenization scripts mentioned above. For scoring Chinese with baseline metrics, we thus pre-processed MT outputs and reference translations with the script tokenizeChinese.py[8] by Shujian Huang, which separates Chinese characters from each other and also from non-Chinese parts.

For computing system-level and segment-level scores, the same scripts were employed as in last year's Metrics Task as well as for generation of hybrid systems from the given hybrid descriptions.

## 3 Results

We discuss system-level results for news task systems in Section 3.1. The segment-level results are in Section 3.2.

### 3.1 System-Level Results

As in previous years, we employ the Pearson correlation ($r$) as the main evaluation measure for system-level metrics. The Pearson correlation is as follows:

$$r = \frac{\sum_{i=1}^{n}(H_i - \overline{H})(M_i - \overline{M})}{\sqrt{\sum_{i=1}^{n}(H_i - \overline{H})^2}\sqrt{\sum_{i=1}^{n}(M_i - \overline{M})^2}} \qquad (2)$$

where $H_i$ are human assessment scores of all systems in a given translation direction, $M_i$ are the corresponding scores as predicted by a given metric. $\overline{H}$ and $\overline{M}$ are their means respectively.

Since some metrics, such as BLEU, for example, aim to achieve a strong positive correlation with human assessment, while error metrics, such as TER aim for a strong negative correlation, after computation of $r$ for metrics, we compare metrics via the absolute value of a given metric's correlation with human assessment.

Table 4 provides the system-level correlations of metrics evaluating translation of newstest2018 into English while Table 5 provides the same for out-of-English language pairs. The underlying texts are part of the WMT18 News Translation test set (newstest2018) and the underlying MT systems are all MT systems participating in the WMT18 News Translation Task with the exception of a single tr-en system not included in the initial human evaluation run.

As recommended by Graham and Baldwin (2014), we employ Williams significance test (Williams, 1959) to identify differences in correlation that are statistically significant. Williams test is a test of significance of a difference in dependent correlations and therefore suitable for evaluation of metrics. Correlations not significantly outperformed by any other metric for the given language pair are highlighted in bold in Tables 4 and 5.

Since pairwise comparisons of metrics may be also of interest, e.g. to learn which metrics

[8] `http://hdl.handle.net/11346/WMT17-TVXH`

| | **cs-en** | **de-en** | **et-en** | **fi-en** | **ru-en** | **tr-en** | **zh-en** |
|---|---|---|---|---|---|---|---|
| $n$ | 5 | 16 | 14 | 9 | 8 | 5 | 14 |
| Correlation | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ |
| BEER | **0.958** | 0.994 | **0.985** | **0.991** | 0.982 | 0.870 | **0.976** |
| BLEND | **0.973** | 0.991 | 0.985 | **0.994** | **0.993** | **0.801** | **0.976** |
| BLEU | **0.970** | 0.971 | **0.986** | 0.973 | 0.979 | **0.657** | **0.978** |
| CDER | **0.972** | 0.980 | **0.990** | 0.984 | 0.980 | **0.664** | **0.982** |
| CharacTER | **0.970** | **0.993** | 0.979 | 0.989 | **0.991** | **0.782** | 0.950 |
| ITER | **0.975** | 0.990 | 0.975 | **0.996** | 0.937 | **0.861** | **0.980** |
| meteor++ | **0.945** | 0.991 | 0.978 | 0.971 | **0.995** | 0.864 | 0.962 |
| NIST | **0.954** | 0.984 | 0.983 | 0.975 | 0.973 | **0.970** | 0.968 |
| PER | **0.970** | 0.985 | **0.983** | **0.993** | 0.967 | 0.159 | 0.931 |
| RUSE | **0.981** | **0.997** | **0.990** | **0.991** | **0.988** | **0.853** | **0.981** |
| TER | **0.950** | 0.970 | **0.990** | 0.968 | 0.970 | 0.533 | **0.975** |
| UHH_TSKM | 0.952 | 0.980 | **0.989** | 0.982 | 0.980 | 0.547 | **0.981** |
| WER | **0.951** | 0.961 | **0.991** | 0.961 | 0.968 | 0.041 | **0.975** |
| YiSi-0 | 0.956 | **0.994** | 0.975 | 0.978 | **0.988** | **0.954** | 0.957 |
| YiSi-1 | **0.950** | 0.992 | 0.979 | 0.973 | **0.991** | **0.958** | 0.951 |
| YiSi-1_srl | **0.965** | **0.995** | 0.981 | 0.977 | **0.992** | **0.869** | 0.962 |

**newstest2018**

Table 4: Absolute Pearson correlation of to-English system-level metrics with DA human assessment in newstest2018; correlations of metrics not significantly outperformed by any other for that language pair are highlighted in bold; ensemble metrics are highlighted in gray.

| | **en-cs** | **en-de** | **en-et** | **en-fi** | **en-ru** | **en-tr** | **en-zh** |
|---|---|---|---|---|---|---|---|
| $n$ | 5 | 16 | 14 | 12 | 9 | 8 | 14 |
| Correlation | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ |
| BEER | **0.992** | **0.991** | **0.980** | **0.961** | **0.988** | **0.965** | 0.928 |
| BLEND | – | – | – | – | **0.988** | – | – |
| BLEU | 0.995 | 0.981 | 0.975 | **0.962** | 0.983 | 0.826 | 0.947 |
| CDER | 0.997 | 0.986 | **0.984** | **0.964** | **0.984** | 0.861 | 0.961 |
| CharacTER | **0.993** | **0.989** | 0.956 | **0.974** | 0.983 | 0.833 | **0.983** |
| ITER | 0.915 | **0.984** | **0.981** | **0.973** | 0.975 | 0.865 | – |
| NIST | **0.999** | 0.986 | **0.983** | 0.949 | **0.990** | 0.902 | 0.950 |
| PER | 0.991 | 0.981 | 0.958 | 0.906 | **0.988** | 0.859 | 0.964 |
| TER | **0.997** | **0.988** | **0.981** | **0.942** | **0.987** | 0.867 | **0.963** |
| WER | **0.997** | **0.986** | **0.981** | **0.945** | 0.985 | 0.853 | 0.957 |
| YiSi-0 | 0.973 | 0.985 | 0.968 | 0.944 | **0.990** | **0.990** | 0.957 |
| YiSi-1 | **0.987** | 0.985 | **0.979** | 0.940 | **0.992** | **0.976** | 0.963 |
| YiSi-1_srl | – | **0.990** | – | – | – | – | 0.952 |

**newstest2018**

Table 5: Absolute Pearson correlation of out-of-English system-level metrics with DA human assessment in newstest2018; correlations of metrics not significantly outperformed by any other for that language pair are highlighted in bold; ensemble metrics are highlighted in gray.

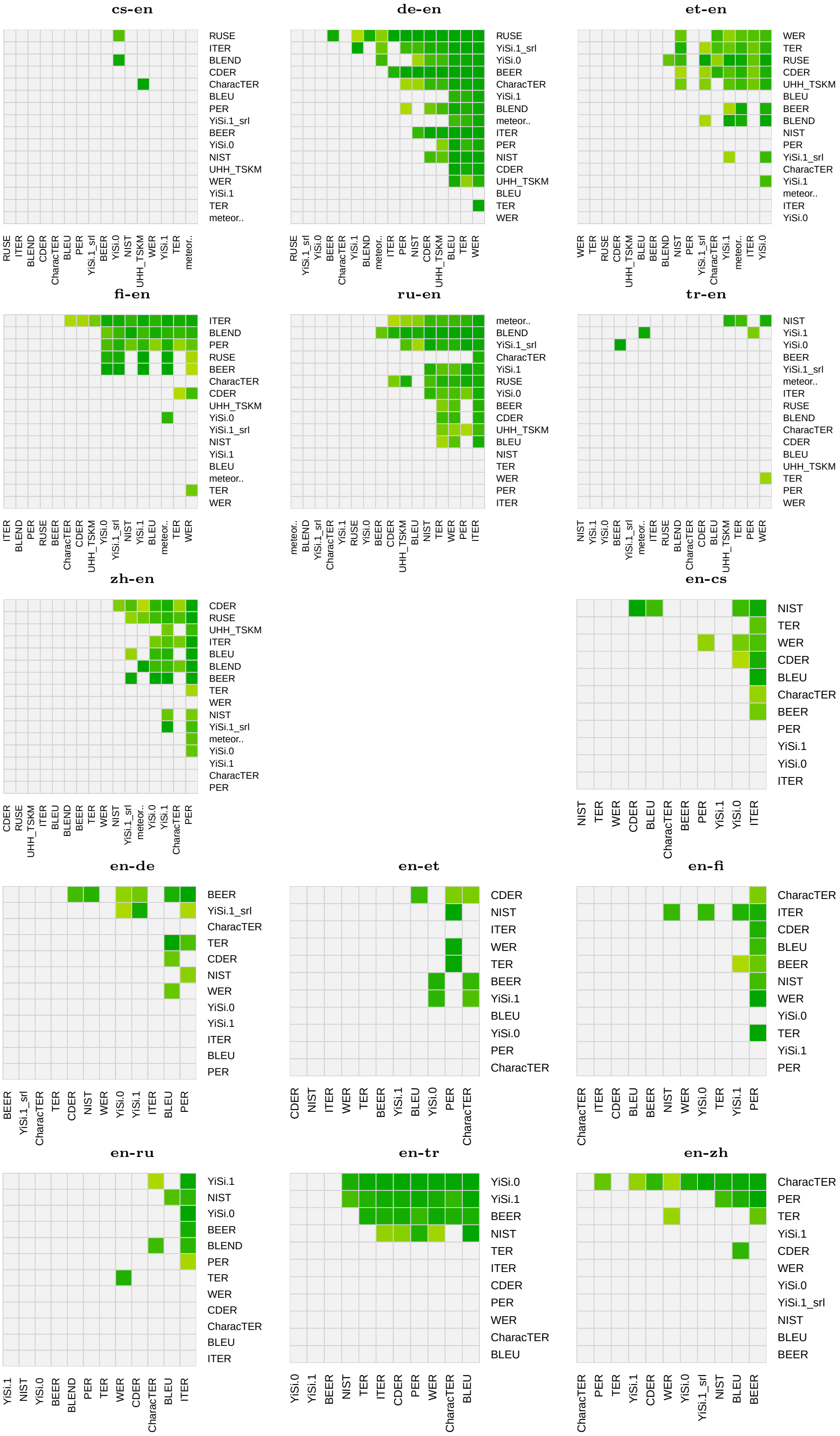


Figure 1: System-level metric significance test results for DA human assessment in newstest2018; green cells denote a statistically significant increase in correlation with human assessment for the metric in a given row over the metric in a given column according to Williams test.

| | **cs-en** | **de-en** | **et-en** | **fi-en** | **ru-en** | **tr-en** | **zh-en** |
|---|---|---|---|---|---|---|---|
| $n$ | 10K | 10K | 10K | 10K | 10K | 10K | 10K |
| Correlation | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ |
| BEER | 0.9497 | 0.9927 | 0.9831 | 0.9824 | 0.9755 | 0.7234 | 0.9677 |
| BLEND | 0.9646 | 0.9904 | 0.9820 | 0.9853 | 0.9865 | 0.7243 | 0.9686 |
| BLEU | 0.9557 | 0.9690 | 0.9812 | 0.9618 | 0.9719 | 0.5862 | 0.9684 |
| CDER | 0.9642 | 0.9797 | 0.9876 | 0.9764 | 0.9739 | 0.5767 | 0.9733 |
| CharacTER | 0.9595 | 0.9919 | 0.9754 | 0.9791 | 0.9841 | 0.6798 | 0.9424 |
| ITER | 0.9656 | 0.9904 | 0.9746 | **0.9885** | 0.9429 | 0.7420 | **0.9780** |
| meteor++ | 0.9367 | 0.9898 | 0.9753 | 0.9621 | **0.9892** | 0.7871 | 0.9541 |
| NIST | 0.9419 | 0.9816 | 0.9804 | 0.9655 | 0.9650 | 0.8622 | 0.9589 |
| PER | 0.9369 | 0.9820 | 0.9782 | 0.9834 | 0.9550 | 0.0433 | 0.9233 |
| RUSE | **0.9736** | **0.9959** | 0.9879 | 0.9829 | 0.9820 | 0.7796 | 0.9734 |
| TER | 0.9419 | 0.9699 | 0.9882 | 0.9599 | 0.9635 | 0.4495 | 0.9670 |
| UHH_TSKM | 0.9429 | 0.9794 | 0.9869 | 0.9738 | 0.9734 | 0.4433 | 0.9717 |
| WER | 0.9420 | 0.9612 | **0.9892** | 0.9534 | 0.9618 | 0.0720 | 0.9667 |
| YiSi-0 | 0.9465 | 0.9925 | 0.9719 | 0.9694 | 0.9817 | 0.8629 | 0.9495 |
| YiSi-1 | 0.9425 | 0.9909 | 0.9758 | 0.9641 | 0.9846 | **0.8810** | 0.9429 |
| YiSi-1_srl | 0.9565 | 0.9940 | 0.9783 | 0.9682 | 0.9860 | 0.7850 | 0.9540 |
| | | | | **newstest2018** Hybrids | | | |

Table 6: Absolute Pearson correlation of to-English system-level metrics with DA human assessment for 10K hybrid super-sampled systems in newstest2018; ensemble metrics are highlighted in gray.

| | **en-cs** | **en-de** | **en-et** | **en-fi** | **en-ru** | **en-tr** | **en-zh** |
|---|---|---|---|---|---|---|---|
| $n$ | 10K | 10K | 10K | 10K | 10K | 10K | 10K |
| Correlation | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ | $\|r\|$ |
| BEER | 0.9903 | **0.9891** | 0.9775 | 0.9587 | 0.9864 | 0.9327 | 0.9251 |
| BLEND | – | – | – | – | 0.9861 | – | – |
| BLEU | 0.9931 | 0.9774 | 0.9706 | 0.9582 | 0.9767 | 0.7963 | 0.9414 |
| CDER | 0.9949 | 0.9842 | 0.9809 | 0.9605 | 0.9821 | 0.8322 | 0.9564 |
| CharacTER | 0.9902 | 0.9862 | 0.9495 | 0.9627 | 0.9814 | 0.7752 | **0.9784** |
| ITER | 0.8649 | 0.9778 | **0.9817** | **0.9664** | 0.9650 | 0.8724 | – |
| NIST | **0.9967** | 0.9839 | 0.9797 | 0.9436 | 0.9877 | 0.8703 | 0.9442 |
| PER | 0.9865 | 0.9787 | 0.9545 | 0.9044 | 0.9862 | 0.8289 | 0.9500 |
| TER | 0.9948 | 0.9861 | 0.9770 | 0.9391 | 0.9845 | 0.8373 | 0.9591 |
| WER | 0.9944 | 0.9842 | 0.9772 | 0.9418 | 0.9829 | 0.8239 | 0.9537 |
| YiSi-0 | 0.9713 | 0.9829 | 0.9648 | 0.9422 | 0.9879 | **0.9530** | 0.9513 |
| YiSi-1 | 0.9851 | 0.9826 | 0.9761 | 0.9384 | **0.9893** | 0.9418 | 0.9572 |
| YiSi-1_srl | – | 0.9881 | – | – | – | – | 0.9479 |
| | | | | **newstest2018** Hybrids | | | |

Table 7: Absolute Pearson correlation of out-of-English system-level metrics with DA human assessment for 10K hybrid super-sampled systems in newstest2018; ensemble metrics are highlighted in gray.

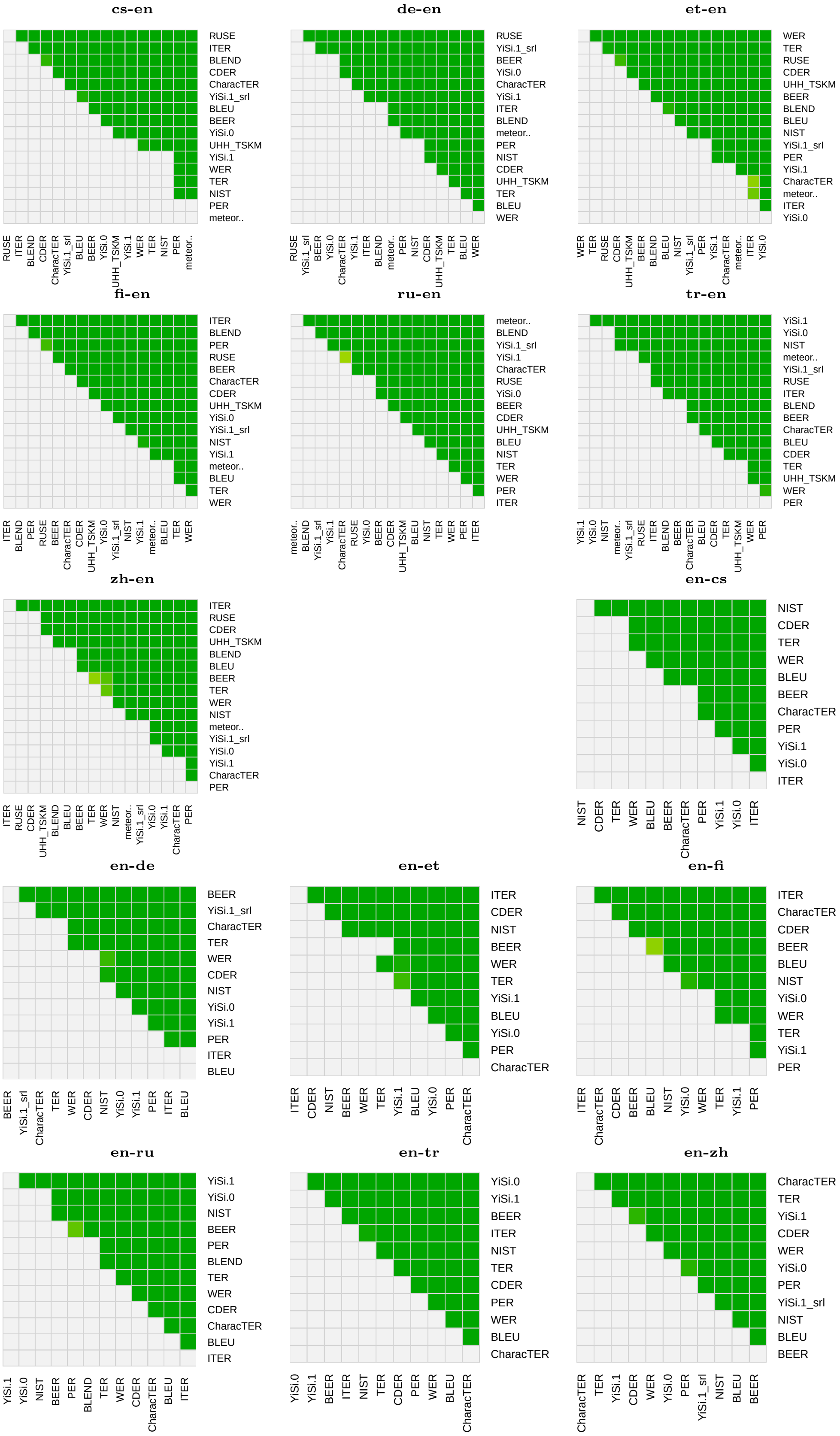


Figure 2: System-level metric significance test results for 10K hybrid systems (DA human evaluation) from newstest2018; green cells denote a statistically significant increase in correlation with human assessment for the metric in a given row over the metric in a given column according to Williams test.

significantly outperform the most widely employed metric BLEU, we include significance test results for every competing pair of metrics including our baseline metrics in Figure 1.

The sample of systems we employ to evaluate metrics is often small, as few as five MT systems for cs-en, for example. This can lead to inconclusive results, as identification of significant differences in correlations of metrics is unlikely at such a small sample size. Furthermore, Williams test takes into account the correlation between each pair of metrics, in addition to the correlation between the metric scores themselves, and this latter correlation increases the likelihood of a significant difference being identified.

To strenghten the conclusions of our evaluation, we include significance test results for large hybrid-super-samples of systems (Graham and Liu, 2016). 10K hybrid systems were created per language pair, with corresponding DA human assessment scores by sampling pairs of systems from WMT18 News Translation Task, creating hybrid systems by randomly selecting each candidate translation from one of the two selected systems. Similar to last year, not all metrics participating in the system-level evaluation submitted metric scores for the large set of hybrid systems. Fortunately, taking a simple average of segment-level scores is the proper aggregation method for almost all metrics this year, so where needed, we provided scores for hybrids ourselves, see Table 2.

Correlations of metric scores with human assessment of the large set of hybrid systems are shown in Tables 6 and 7, where again metrics not significantly outperformed by any other are highlighted in bold. Figure 2 then provides significance test results for hybrid super-sampled correlations for all pairs of competing metrics for a given language pair.

### 3.2 Segment-Level Results

Segment-level evaluation relies on the manual judgements collected in the News Translation Task evaluation. This year, we were unable to follow the methodology outlined in Graham et al. (2015) for evaluation of segment-level metrics because the sampling of sentences did not provide sufficient number of assessments of the same segment. We therefore convert pairs of DA scores for competing translations to daRR better/worse preferences and employ a Kendall's Tau formulation as described in Section 2.3.2.

Results of the segment-level human evaluation for translations sampled from the News Translation Task are shown in Tables 8 and 9, where metric correlations not significantly outperformed by any other metric are highlighted in bold. Head-to-head significance test results for differences in metric performance are included in Figure 3.

## 4 Discussion

### 4.1 Obtaining Human Judgements

Human data was collected in the usual way, a portion via crowd-sourcing and the remaining from researchers who mainly committed their time contribution to the manual evaluation as they had submitted a system in that language pair. Evaluation of translations employed the DA set-up and it again successfully acquired sufficient judgments to evaluate systems. As in the previous years, hybrid super-sampling proved very effective and allowed to obtain conclusive results of system-level evaluation even for language pairs where as few as 5 MT systems participated. We should however note that hybrid systems are constructed by randomly mixing sentences coming from different MT systems. As soon as document-level evaluation becomes relevant (which we anticipate in the next evaluation campaign already), this style of hybridization is susceptible to breaking cross-sentence references in MT outputs and may no longer be applicable.

In the case of segment-level evaluation, the optimal human evaluation data was unfortunately not available due to resource constraints. Conversion of document-level data held as a substitute for segment-level DA scores. These scores are however not optimal for evaluation of segment-level metrics and we would like to return to DA's standard segment-level evaluation in future, where a minimum of 15 human judgments of translation quality are collected per translation and combined to get highly accurate scores for translations.

| | **cs-en** | **de-en** | **et-en** | **fi-en** | **ru-en** | **tr-en** | **zh-en** |
|---|---|---|---|---|---|---|---|
| Human Evaluation | daRR | daRR | daRR | daRR | daRR | daRR | daRR |
| $n$ | 5,110 | 77,811 | 56,721 | 15,648 | 10,404 | 8,525 | 33,357 |
| Correlation | $\tau$ | $\tau$ | $\tau$ | $\tau$ | $\tau$ | $\tau$ | $\tau$ |
| BEER | 0.295 | 0.481 | 0.341 | 0.232 | **0.288** | **0.229** | **0.214** |
| BLEND | **0.322** | **0.492** | 0.354 | 0.226 | **0.290** | **0.232** | **0.217** |
| CharacTER | 0.256 | 0.450 | 0.286 | 0.185 | 0.244 | 0.172 | **0.202** |
| ITER | 0.198 | 0.396 | 0.235 | 0.128 | 0.139 | −0.029 | 0.144 |
| meteor++ | 0.270 | 0.457 | 0.329 | 0.207 | 0.253 | 0.204 | 0.179 |
| RUSE | **0.347** | **0.498** | **0.368** | **0.273** | **0.311** | **0.259** | **0.218** |
| sentBLEU | 0.233 | 0.415 | 0.285 | 0.154 | 0.228 | 0.145 | 0.178 |
| UHH_TSKM | 0.274 | 0.436 | 0.300 | 0.168 | 0.235 | 0.154 | 0.151 |
| YiSi-0 | 0.301 | 0.474 | 0.330 | 0.225 | **0.294** | 0.215 | **0.205** |
| YiSi-1 | **0.319** | 0.488 | 0.351 | 0.231 | **0.300** | **0.234** | **0.211** |
| YiSi-1_srl | **0.317** | 0.483 | 0.345 | 0.237 | **0.306** | **0.233** | **0.209** |

**newstest2018**

Table 8: Segment-level metric results for to-English language pairs in newstest2018: absolute Kendall's Tau formulation of segment-level metric scores with DA scores; correlations of metrics not significantly outperformed by any other for that language pair are highlighted in bold; ensemble metrics are highlighted in gray.

| | **en-cs** | **en-de** | **en-et** | **en-fi** | **en-ru** | **en-tr** | **en-zh** |
|---|---|---|---|---|---|---|---|
| Human Evaluation | daRR | daRR | daRR | daRR | daRR | daRR | daRR |
| $n$ | 5,413 | 19,711 | 32,202 | 9,809 | 22,181 | 1,358 | 28,602 |
| Correlation | $\tau$ | $\tau$ | $\tau$ | $\tau$ | $\tau$ | $\tau$ | $\tau$ |
| BEER | **0.518** | **0.686** | **0.558** | **0.511** | **0.403** | **0.374** | 0.302 |
| BLEND | – | – | – | – | **0.394** | – | – |
| CharacTER | 0.414 | 0.604 | 0.464 | 0.403 | 0.352 | **0.404** | **0.313** |
| ITER | 0.333 | 0.610 | 0.392 | 0.311 | 0.291 | 0.236 | – |
| sentBLEU | 0.389 | 0.620 | 0.414 | 0.355 | 0.330 | 0.261 | **0.311** |
| YiSi-0 | 0.471 | 0.661 | 0.531 | 0.464 | **0.394** | **0.376** | **0.318** |
| YiSi-1 | **0.496** | **0.691** | **0.546** | **0.504** | **0.407** | **0.418** | **0.323** |
| YiSi-1_srl | – | **0.696** | – | – | – | – | **0.310** |

**newstest2018**

Table 9: Segment-level metric results for out-of-English language pairs in newstest2018: absolute Kendall's Tau formulation of segment-level metric scores with DA scores; correlations of metrics not significantly outperformed by any other for that language pair are highlighted in bold; ensemble metrics are highlighted in gray.

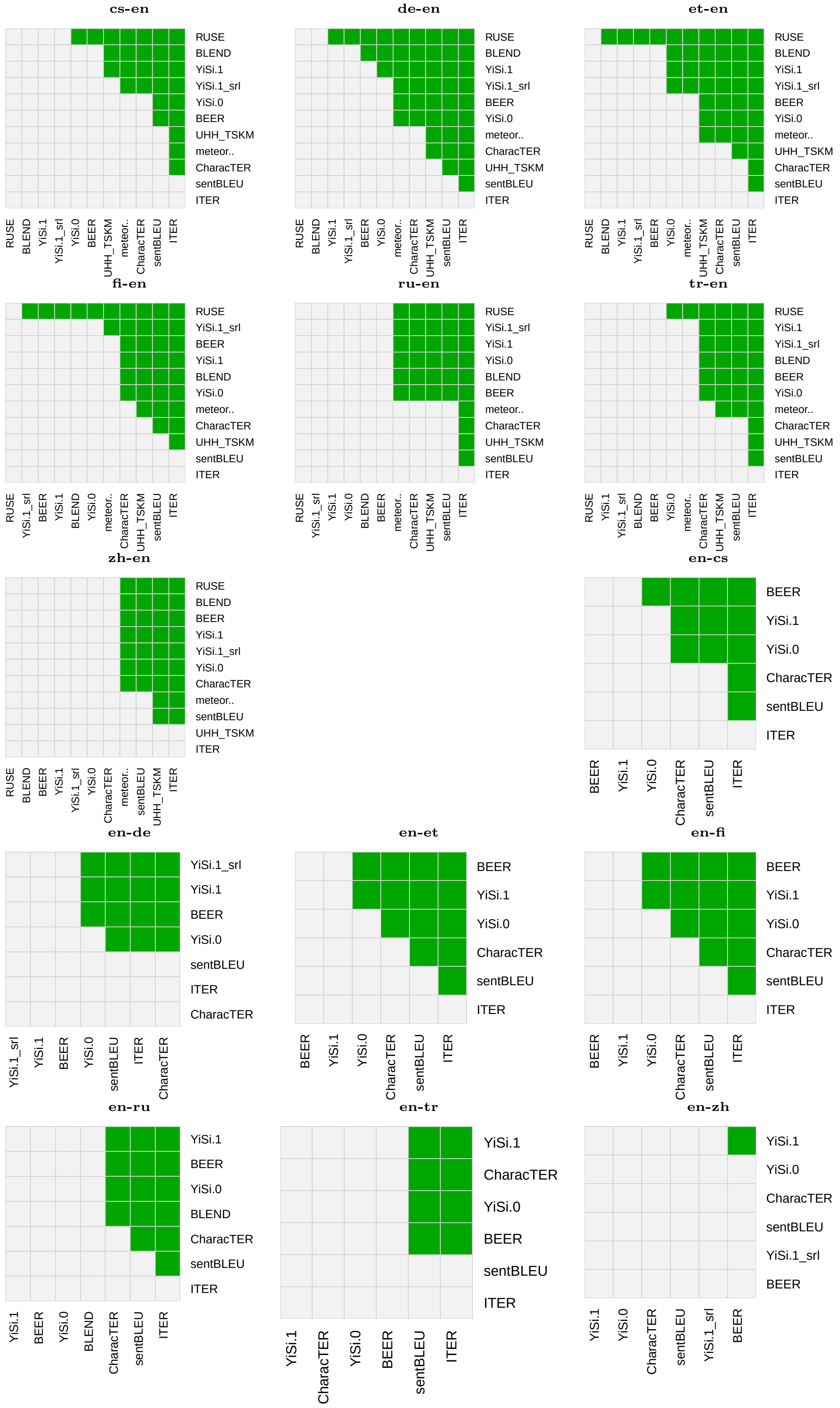


Figure 3: DARR segment-level metric significance test results for all language pairs (newstest2018): Green cells denote a significant win for the metric in a given row over the metric in a given column according bootstrap resampling.

### 4.2 Overall Metric Performance

As always, the observed performance of metrics depends on the underlying texts and systems that participate in the News Translation Task. Two new metrics, RUSE and YiSi stand out as metrics that achieve highest correlation in the system level evaluation in more than one language pair according to the hybrid evaluation, and perform great across all their language pairs on average. ITER also performs very well in en-et, en-fi, zh-en and several other languages but fails for en-ru and en-cs, which drags its overall performance down.

Both YiSi and RUSE are based on neural networks (YiSi via word and phrase embeddings, RUSE via sentence embeddings). This is a new trend compared to the last year evaluation where the best performance was reached by character-level (not deep) metrics BEER, chrF (and its variants) and CharacTer.

It is however important to note that the results of performance agreggated over language pairs are not particularly stable across years. In the last year's evaluation, NIST seemed worse than TER. The overall results is the opposite this year and NIST even ranks slightly better than RUSE in terms of average system-level correlation across languages.

Overall, the reported figures confirm the observation from the past years that system-level metrics can achieve correlations above 0.9 but even the best ones can fall to 0.7 or 0.8 for some language pairs. Kendall's Tau achieved by segment-level metrics are generally lower, in the range of 0.25–0.4. The best metrics in their best language pairs can reach up to 0.69 of segment-level correlations with humans. This capping could be possibly in part attributed to the sub-optimal human evaluation data, DA judgements converted to relative ranking.

Two metrics that stand out as performing consistently well are RUSE for evaluation of into-English translation and YiSi-1* for out-of-English. Overall, YiSi*, BEER, CharacTER, RUSE, and BLEND (in this order) outperform sentBLEU.

All of the "winners" in this years campaign are publicly available, which is very good for their prospective wider adoption. If participants could put the additional effort of adding their code to Moses scorer, this would guarantee their long-term inclusion in the Metrics Task.

## 5 Conclusion

This paper summarizes the results of WMT18 shared task in machine translation evaluation, the Metrics Shared Task. Participating metrics were evaluated in terms of their correlation with human judgment at the level of the whole test set (system-level evaluation), as well as at the level of individual sentences (segment-level evaluation). For the former, best metrics reach over 0.95 Pearson correlation or better across several language pairs. Correlations varied more than usual between 0.2 and 0.7 in terms of segment-level metrics Kendall's $\tau$ results.

## Acknowledgments

Results in this shared task would not be possible without tight collaboration with organizers of the WMT News Translation Task.

This study was supported in parts by the grants 18-24210S of the Czech Science Foundation, ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Dublin City University funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund, and Charles University Research Programme "Progres" Q18+Q48.

## References

Ondřej Bojar, Miloš Ercegovčević, Martin Popel, and Omar Zaidan. 2011. A Grain of Salt for the WMT Manual Evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, Edinburgh, Scotland, July. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Barry Haddow, Philipp Koehn, Matt Post, and Lucia Specia. 2016. Ten Years of WMT Evaluation Campaigns: Lessons Learnt. In *Proceedings of the LREC 2016 Workshop "Translation Evaluation – From Fragmented Tools and Data Sets to an Integrated Ecosystem"*, pages 27–34, Portorose, Slovenia, 5.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark, September. Association for Computational Linguistics.

Ondřej Bojar, Jiří Mírovský, Kateřina Rysová, and Magdaléna Rysová. 2018. Evald reference-less discourse evaluation for wmt18. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels, October. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Melania Duma and Wolfgang Menzel. 2017. UHH submission to the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark, September. Association for Computational Linguistics.

Yvette Graham and Timothy Baldwin. 2014. Testing for Significance of Increased Correlation with Human Judgment. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 172–176, Doha, Qatar, October. Association for Computational Linguistics.

Yvette Graham and Qun Liu. 2016. Achieving Accurate Conclusions in Evaluation of Automatic Machine Translation Metrics. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, CA. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous Measurement Scales in Human Evaluation of Machine Translation. In *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2014a. Is Machine Translation Getting Better over Time? In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 443–451, Gothenburg, Sweden, April. Association for Computational Linguistics.

Yvette Graham, Nitika Mathur, and Timothy Baldwin. 2014b. Randomized significance tests in machine translation. In *Proceedings of the ACL 2014 Ninth Workshop on Statistical Machine Translation*, pages 266–274. Association for Computational Linguistics.

Yvette Graham, Nitika Mathur, and Timothy Baldwin. 2015. Accurate Evaluation of Segment-level Machine Translation Metrics. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, Denver, Colorado.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, FirstView:1–28, 1.

Yinuo Guo, Chong Ruan, and Junfeng Hu. 2018. Meteor++: Incorporating copy knowledge into machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels, October. Association for Computational Linguistics.

Philipp Koehn and Christof Monz. 2006. Manual and Automatic Evaluation of Machine Translation Between European Languages. In *Proceedings of the Workshop on Statistical Machine Translation*, StatMT '06, pages 102–121, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Gregor Leusch, Nicola Ueffing, and Hermann Ney. 2006. CDER: Efficient MT Evaluation Using Block Movements. In *In Proceedings of EACL*, pages 241–248.

Chi-kiu Lo. 2018. The NRC metric submission to the WMT18 metric and parallel corpus filtering shared task. In *Arxiv*.

Qingsong Ma, Yvette Graham, Shugen Wang, and Qun Liu. 2017. Blend: a novel combined MT metric based on direct assessment — casict-dcu submission to WMT17 metrics task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark, September. Association for Computational Linguistics.

Matouš Macháček and Ondřej Bojar. 2014. Results of the WMT14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, Baltimore, MD, USA. Association for Computational Linguistics.

Matouš Macháček and Ondřej Bojar. 2013. Results of the WMT13 Metrics Shared Task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 45–51, Sofia, Bulgaria, August. Association for Computational Linguistics.

Joybrata Panja and Sudip Kumar Naskar. 2018. Iter: Improving translation edit rate through optimizable edit costs. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels, October. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisboa, Portugal, September. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark, September. Association for Computational Linguistics.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. Ruse: Regressor using sentence embeddings for automatic machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels, October. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Miloš Stanojević and Khalil Sima'an. 2015. BEER 1.1: ILLC UvA submission to metrics and tuning task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisboa, Portugal, September. Association for Computational Linguistics.

Yolanda Vazquez-Alvarez and Mark Huckvale. 2002. The reliability of the ITU-t p.85 standard for the evaluation of text-to-speech systems. In *Proc. of ICSLP - INTERSPEECH*.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016a. Character: Translation edit rate on character level. In *ACL 2016 First Conference on Machine Translation*, pages 505–510, Berlin, Germany, August.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016b. CharacTer: Translation Edit Rate on Character Level. In *Proceedings of the First Conference on Machine Translation*, Berlin, Germany, August. Association for Computational Linguistics.

Evan James Williams. 1959. *Regression analysis*, volume 14. Wiley New York.

# Findings of the WMT 2018 Shared Task on Quality Estimation

**Lucia Specia** and **Frédéric Blain**
Department of Computer Science
University of Sheffield, UK
{l.specia,f.blain}@sheffield.ac.uk

**Varvara Logacheva**
Neural Networks and Deep Learning Lab
MIPT, Moscow, Russia
logacheva.vk@mipt.ru

**Ramón F. Astudillo**
L2F, INESC-ID-Lisboa
Lisbon, Portugal
ramon@astudillo.com

**André Martins**
Unbabel & Instituto de Telecomunicações
Lisbon, Portugal
andre.martins@unbabel.com

## Abstract

We report the results of the WMT18 shared task on Quality Estimation, i.e. the task of predicting the quality of the output of machine translation systems at various granularity levels: word, phrase, sentence and document. This year we include four language pairs, three text domains, and translations produced by both statistical and neural machine translation systems. Participating teams from ten institutions submitted a variety of systems to different task variants and language pairs.

## 1 Introduction

This shared task builds on its previous six editions to further examine automatic methods for estimating the quality of machine translation (MT) output at run-time, without the use of reference translations. It includes the (sub)tasks of word-level, phrase-level, sentence-level and document-level estimation. In addition to advancing the state of the art at all prediction levels, our goals include:

- To study the performance of quality estimation approaches on the output of neural MT systems. We do so by providing datasets for two language pairs where source segments were translated by both statistical phrase-based and neural MT systems.
- To study the predictability of missing words in the MT output. To do so, for the first time we provide data annotated for such errors at training time.
- To study the predictability of source words that lead to errors in the MT output. To do so, for the first time we provide source segments annotated for such errors at the word level.
- To study the effectiveness of manually assigned labels for phrases. For that we provide a dataset where each phrase was annotated by human translators.
- To investigate the utility of detailed information logged during post-editing. We do so by providing post-editing time, keystrokes, as well as post-editor ID.
- To study quality prediction for documents from errors annotated at word-level with added severity judgements. This is done using a new corpus manually annotated with a fine-grained error taxonomy, from which document-level scores are derived.

This year's shared task provides new training and test datasets for all tasks, and allows participants to explore any additional data and resources deemed relevant. Tasks make use of large datasets produced either from post-editions or annotations by professional translators, or from direct human annotations. The following text domains are available for different languages and tasks: information technology (IT), life sciences, and product title and descriptions on sports and outdoor activities. In-house statistical and neural MT systems were built to produce translations for the two first domains, while an online system was used for the third domain.

The four tasks are defined as follows: Task 1 aims at predicting post-editing effort at sentence level (Section 5); Task 2 aims at predicting words that need editing, as well as missing words and incorrect source words (Section 6); Task 3 aims at predicting phrases that need editing, as well as missing phrases and incorrect source phrases (Section 7); and Task 4 (Section 8) aims at predicting a score for an entire document as a function of the proportion of incorrect words in such a document, weighted by the severity of the different errors.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 689–709
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64078

Five datasets and language pairs are used for different tasks (Section 4): English-German (Tasks 1, 2) and English-Czech (Tasks 1, 2) on the IT domain, English-Latvian (Tasks 1, 2) and German-English (Tasks 1, 2, 3), both on the life sciences domain, English-French (Task 4) with product titles and descriptions within the sports and outdoor activities domain.

Participants are provided with a baseline set of features for each task, and a software package to extract these and other quality estimation features, and perform model learning (Section 2). Participants (Section 3) could submit up to two systems for each task and language pair. A discussion on the main goals and findings from this year's task is given in Section 9.

## 2 Baseline systems

**Sentence-level baseline system:** For Task 1, QuEst++[1] (Specia et al., 2015) was used to extract 17 MT system-independent features from the source and translation (target) files and parallel corpora:

- Number of tokens in the source and target sentences.
- Average source token length.
- Average number of occurrences of the target word within the target sentence.
- Number of punctuation marks in source and target sentences.
- Language model (LM) probability of source and target sentences based on models built using the source or target sides of the parallel corpus used to train the SMT system.
- Average number of translations per source word in the sentence as given by the IBM model 1 extracted using the SMT parallel corpus, and thresholded such that $P(t|s) > 0.2$ or $P(t|s) > 0.01$.
- Percentage of unigrams, bigrams and trigrams in frequency quartiles 1 (lower frequency words) and 4 (higher frequency words) in the source language extracted from the source side of the SMT parallel corpus.
- Percentage of unigrams in the source sentence seen in the source side of the SMT parallel corps.

These features were used to train a Support Vector Regression (SVR) algorithm using a Radial Basis Function (RBF) kernel within the SCIKIT-LEARN toolkit.[2] The $\gamma$, $\epsilon$ and $C$ parameters were optimised via grid search with 5-fold cross validation on the training set, resulting in $\gamma$=0.01, $\epsilon$ = 0.0825, $C$ = 20. This baseline system has been consistently used as the baseline system for all editions of the sentence-level task (Callison-Burch et al., 2012; Bojar et al., 2013, 2014, 2015, 2016, 2017), and has proved strong enough for predicting various forms of post-editing effort across a range of language pairs and text domains for statistical MT systems. This year it is also benchmarked on neural MT outputs.

**Word-level baseline system:** For Task 2, the baseline features were extracted with the MARMOT tool (Logacheva et al., 2016). These are 28 features that have been deemed the most informative in previous research on word-level QE, mostly inspired by (Luong et al., 2014). This is the same baseline system used in WMT17:

- Word count in the source and target sentences, and source and target token count ratio. Although these features are sentence-level (i.e. their values will be the same for all words in a sentence), the length of a sentence might influence the probability of a word being wrong.
- Target token, its left and right contexts of one word.
- Source word aligned to the target token, its left and right contexts of one word. The alignments were given by the SMT system that produced the automatic translations.
- Boolean dictionary features: target token is a stop word, a punctuation mark, a proper noun, or a number.
- Target language model features:
  - The order of the highest order ngram which starts and end with the target token.
  - The order of the highest order ngram which starts and ends with the source token.
  - The part-of-speech (POS) tags of the target and source tokens.
  - Backoff behaviour of the ngrams $(t_{i-2}, t_{i-1}, t_i)$, $(t_{i-1}, t_i, t_{i+1})$, $(t_i, t_{i+1}, t_{i+2})$, where $t_i$ is the target

[1] https://github.com/ghpaetzold/questplusplus

[2] http://scikit-learn.org/

token (backoff behaviour is computed as described by (2011)).

In addition to that, six new features were included which contain combinations of other features, and which proved useful in (Kreutzer et al., 2015; Martins et al., 2016):

- Target word + left context.
- Target word + right context.
- Target word + aligned source word.
- POS of target word + POS of aligned source word.
- Target word + left context + source word.
- Target word + right context + source word.

The baseline system models the task as a sequence prediction problem using the Linear-Chain Conditional Random Fields (CRF) algorithm within the CRFSuite tool.[3] The model was trained using passive-aggressive optimisation algorithm.

We note that this baseline system was only used to predict OK/BAD classes for existing words in the MT output. No baseline system was provided for predicting missing words or erroneous source words.

**Phrase-level baseline system:** The phrase-level system is identical to the one used in last year's shared task. The phrase-level features were also extracted with MARMOT, but they are different from the word-level features. They are based on the sentence-level features in QUEST++.[4] These are the so-called "black-box" features – features that do not use the internal information from the MT system. The baseline uses the following 72 features:

- Source phrase frequency features:
  - average frequency of ngrams (unigrams, bigrams, trigrams) in different quartiles of frequency (the low and high frequency ngrams) in the source side of the SMT parallel corpus.
  - percentage of distinct source ngrams (unigrams, bigrams, trigrams) seen in the source side of the SMT parallel corpus.
- Translation probability features:
  - average number of translations per source word in the phrase as given by the IBM model 1 extracted using the SMT parallel corpus (with different translation probability thresholds: 0.01, 0.05, 0.1, 0.2, 0.5).
  - average number of translations per source word in the phrase as given by the IBM model 1 extracted using the SMT parallel corpus (with different translation probability thresholds: 0.01, 0.05, 0.1, 0.2, 0.5) weighted by the frequency of each word in the source side of the parallel SMT corpus.
- Punctuation features:
  - difference between numbers of various punctuation marks (periods, commas, colons, semicolons, question and exclamation marks) in the source and the target phrases.
  - difference between numbers of various punctuation marks normalised by the length of the target phrase.
  - percentage of punctuation marks in the target or source phrases.
- Language model features:
  - log probability of the source or target phrases based on models built using the source or target sides of the parallel corpus used to train the SMT system.
  - perplexity of the source and the target phrases using the same models as above.
- Phrase statistics:
  - lengths of the source or target phrases.
  - ratio between the source and target phrase lengths.
  - average length of tokens in source or target phrases.
  - average occurrence of target word within the target phrase.
- Alignment features:
  - number of unaligned target words, using the word alignment provided by the SMT decoder.
  - number of target words aligned to more than one source word.

[3] http://www.chokkan.org/software/crfsuite/

[4] http://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox

- average number of alignments per word in the target phrase.

- Part-of-speech features:
  - percentage of content words in the source or target phrases.
  - percentage of words of a particular part of speech tag (verb, noun, pronoun) in the source or target phrases.
  - ratio of numbers of words of a particular part of speech (verb, noun, pronoun) between the source and target phrases.
  - percentage of numbers and alphanumeric tokens in the source or target phrases.
  - ratio between the percentage of numbers and alphanumeric tokens in the source and target phrases.

Analogously to the baseline word-level system, we treat phrase-level QE as a sequence labelling task, and model it using CRF from the CRFSuite toolkit and the passive-aggressive optimisation algorithm.

Once more, this baseline system was only used to predict OK/BAD classes for existing phrases in the MT output. No baseline system was provided for predicting missing phrases or erroneous source phrases.

## 3 Participants

Table 1 lists all participating teams submitting systems to any of the tasks. Each team was allowed up to two submissions for each task variant and language pair. In the descriptions below, participation in specific tasks is denoted by a task identifier (T1 = Task 1, T2 = Task 2, T3 = Task 3, T4 = Task 4).

CMU-LTI (T2):

The CMU-LTI team proposes a Contextual Encoding model for QE. The model consists in three major parts that encode the local and global context information for each target word. The first part uses an embedding layer to represent words and their POS tags in both languages. The second part leverages a one-dimensional convolution layer to integrate local context information for each target word. The third part applies a stack of feed-forward and recurrent neural networks to further encode the global context in the sentence before making the predictions. Syntactic features, such as ngrams, are then integrated to the final feed-forward layer in the neural model. This model achieves competitive results on the English-Czech and English-Latvian word-level QE task.

JU-USAAR (T2):

JU-USAAR presents two approaches to word-level QE: (i) a Bag-of-Words (BoW) model, and (ii) a Paragraph Vector (Doc2Vec) model (Le and Mikolov, 2014). In the BoW model, bag-of-words are prepared from source sentences for each target word appearing in both the MT and PE output in the training data. For every target word appearing in the MT output in the development set, the cosine similarity between the corresponding source sentence and the bag-of-words for the same target word is computed. From this result, a threshold (for the target word) is defined above which the word is retained (i.e., considered 'OK'). In the Doc2Vec-based approach, for each target word appearing in both MT and PE output in the training data, two document vectors are prepared from (i) the corresponding source sentences and (ii) the bag-of-words (as in the BoW model) of the target word. Next, the similarity between these two document vectors for every target word is computed. From the Doc2Vec similarity score and the corresponding PE decision (i.e., whether or not the target word is retained in the PE in the training dataset), a system level threshold is defined. For the test set sentences, if the Doc2Vec similarity score for a target word exceeds this threshold value, then the target word labelled as 'OK', otherwise it is labelled as 'BAD'.

MEQ (T1)

The Vicomtech team submitted two approaches. uMQE is an unsupervised minimalist approach based on two simple measures of accuracy and fluency, respectively. Accuracy is computed via overlapping lexical translation bags of words, with a set expansion mechanism based on longest common prefixes and surface-defined named entities. Fluency is computed by taking the inverse of cross-entropy, according to an in-domain language model. Both measures are combined

| ID | Participating team |
|---:|---|
| CMU-LTI | Carnegie Melon University, US (Hu et al., 2018) |
| JU-USAAR | Jadavpur University, India & University of Saarland, Germany (Basu et al., 2018) |
| MQE | Vicomtech, Spain (Etchegoyhen et al., 2018) |
| QEbrain | Alibaba Group Inc, US (Wang et al., 2018) |
| RTM | Referential Translation Machines, Turkey (Biçici, 2018) |
| SHEF | University of Sheffield, UK (Ive et al., 2018b) |
| TSKQE | University of Hamburg (Duma and Menzel, 2018) |
| UAlacant | University of Alacant, Spain (Sánchez-Martíínez et al., 2018) |
| UNQE | Jiangxi Normal University, China |
| UTartu | University of Tartu, Estonia (Yankovskaya et al., 2018) |

Table 1: Participants in the WMT18 Quality Estimation shared task.

via simple arithmetic means on rescaled values, i.e., no machine learning is used. Since it is unsupervised, the method can only be meaningfully evaluated on the ranking task. sMQE uses the same two features as uMQE, but with supervision. A Support Vector Regressor based on these two features is trained on the available data and used to predict QE scores.

QEbrain (T1, T2):

QE brain uses a conditional target language model as a robust feature extractor with a novel bidirectional transformer which is pretrained on a large parallel corpus filtered to contain "in-domain like" sentences. For QE inference, the feature extraction model can produce not only the high-level joint latent semantic representation between the source and the machine translation, but real-valued measurements of possible erroneous tokens based on the prior knowledge learned from the parallel data. More specifically, it uses the multi-head self-attention mechanism and transformer neural networks (Vaswani et al., 2017) to build the language model. It contains one transformer encoder for the source and a bidirectional transformer encoder for the target. After the feature extraction model is trained, the features are extracted and combined with human-crafted features from the QE baseline system and fed into a Bi-LSTM predictive model for QE. A greedy ensemble selection method is used to decrease the individual model errors and increase model diversity. The bi-LSTM QE model is trained on the official QE data plus artificially generated data and fine-tuned with only the official WMT18 QE data.

RTM (T1, T2, T3, T4):

These submissions build on the previous year's Referential Translation Machine (RTM) approach (Biçici, 2017). RTMs predict data translation between the instances in the training set and the test set using interpretants, data close to the task instances. Interpretants provide context for the prediction task and are used during the derivation of the features measuring the closeness of the test sentences to the training data, the difficulty of translating them, and to identify translation acts between any two data sets for building prediction models. Task-specific quality prediction RTM models are built using the WMT News translation task corpora, taking MT models as a black-box and predicting translation scores independently on the MT model. Multiple machine learning techniques are used and averaged based on their training set performance for label prediction. For sequence classification tasks (T2 and T3), Global Linear Models with dynamic learning (Bicici, 2013) are used.

SHEF (T1, T2, T3, T4):

SHEF submitted two systems per task variant: SHEF-PT and SHEF-bRNN. SHEF-PT is based on a re-implementation of the POSTECH system of (Kim et al., 2017), SHEF-bRNN uses a bidirectional recurrent neural network (bRNN) (Ive et al., 2018a). PT systems are pre-trained using in-domain corpora provided by the organisers. bRNN systems uses two encoders to learn representations of <source, MT> sentence pairs. These representations are used directly to make word-level predictions. A weighted sum over word representations as defined by an attention mechanism is used to make sentence-level predictions. For phrase-level, a standard attention-based neural MT architecture is used. Different parts of the source sentence are attended to produce MT word vectors. Phrase-level predictions are based on representations computed as the sum of their word vectors. For predicting source tags, the source and MT inputs to the models are swapped. The document-level architecture wraps the sentence-level PT and bRNN architectures. PT systems are pre-trained using either additional in-domain or out-of-domain Europarl data. For the multi-task learning system (SHEF-mtl), weights of sentence-level modules are pre-trained to predict sentence MQM scores.

TSKQE (T1):

The TSKQE submissions represent an extension over the previous UHH-STK submissions to the WMT17 QE shared task, which combine the power of sequence and tree kernels applied on source segments, candidate translation and back-translations of the MT output into the source language. In addition, in order to predict the HTER scores, one of the current submissions also explores pseudo-references, which were obtained by translating the source sentences into the target language using an online MT system. The sequence kernels were applied on the tokenised data, while tree kernels were applied to dependency trees.

UAlacant (T1, T2):

The UAlacant submissions use phrase tables from OPUS[5] and a two hidden layer feed-forward neural network for word-level MT QE. Phrase tables are used to extract features for each word and gap in the machine-translated segment for which quality is estimated. These features are then used together with the baseline features for predicting the need of a deletion or an insertion. The neural network takes as input not only the features for the word and the gap on which a decision is to be made, but also the features of the surrounding gaps and words in a sliding-window fashion within a context window of size three. The predictions made at the word level allow to obtain an approximate HTER score which is used for the submissions to the sentence-level task.

UNQE (T1):

The UNQE submissions employ the unified neural network architecture (UNQE) for sentence-level QE tasks (Li et al., 2018). The approach combines a bidirectional RNN encoder-decoder with attention mechanism sub-network and an RNN into a single large neural network, which extracts the quality vectors of the translation outputs through the bidirectional RNN encoder-decoder, and predicts the HTER value of the translation output by RNN. The input text goes through tokenisation, true casing and sub-word unit segmentation. The models are pre-trained with a large parallel bilingual corpus and fine-tuned with the training data of the sentence-level QE share task. The results submitted are averages of the predicted HTER scores under different dimension settings.

UTartu (T1):

UTartu proposes two methods for the sentence-level task. The first method uses attention weights of a neural MT system applied to each sentence pair to compute the probability of the output sentence under the model (forced-decoding). The confidence of the model is computed via metrics of average entropy of the attention weights per each input/output token. The second method computes the `bleu2vec` metric, which extends BLEU with token or n-gram embeddings, but here the metric is made cross-lingual by means of an unsupervised cross-lingual mapping between the source and target language embedding spaces. Three versions of the resulting metric are used: one based on 3-grams, one with tokens (unigrams) and one with byte-pair encoded sub-words (also unigrams). Both submissions use the 17 standard black-box features implemented in QuEst. QuEst+Attention combines them with the first approach and QuEst+Att+CrEmb3 combines QuEst and both approaches together.

[5] http://opus.nlpl.eu/

## 4 Datasets

This year we further expand the datasets used in WMT17 by adding: more instances (see Table 2), more languages (four language pairs), more MT architectures (neural and statistical MT), and different types of annotation (manual and extracted from manual post-editing). In addition, new data was collected and provided for Task 4, on a fifth language pair and third text domain.

### 4.1 Tasks 1 and 2

The initial data was collected as part of the QT21 project[6] and is fully described in (Specia et al., 2017). However, for all language pairs and MT system types, we filtered this data to remove most cases with no edits performed. A skewed distribution towards good quality translations has been shown to be a problem in previous years, and is even more critical with NMT outputs, where up to about half of the MT sentences require no post-editing at all. We kept only a small proportion of HTER=0 sentences in training, development and test sets.

The structure used for the data has been the same since WMT15. Each data instance consists of (i) a source sentence, (ii) its automatic translation into the target language, (iii) the manually post-edited version of the automatic translation, (iv) one or more post-editing effort scores as labels. Professional post-edits are used to extract labels for the two different levels of granularity (word and sentence). Table 2 shows the various resulting datasets for English-German (EN-DE), German-English (DE-EN), English-Latvian (EN-LV) and English-Czech (EN-CS), for both statistical (SMT) and neural (NMT) outputs.

English-German and English-Czech sentences are from the IT domain and were translated by an in-house phrase-based SMT system, and in addition by an in-house encoder-decoder attention-based NMT system for English-German. We note that the original dataset sizes for these languages was 30,000 sentences in total for English-German (per MT system type), and 45,000 for English-Czech. The large reduction in the NMT version of the English-German data indicates the high quality of the NMT system used to produce these sentences: a large number of sentences was filtered out for having undergone no edits by translators.

German-English and English-Latvian sentences are from the life sciences (pharmaceutical) domain and were translated by an in-house phrase-based SMT system, and in addition by an in-house encoder-decoder attention-based NMT system for English-Latvian. The original sentence numbers for these languages were 45,000 and 20,738, respectively (per MT system type).

### 4.2 Task 3

This task uses a subset of the German-English SMT data from Task 1 (5,921 sentences for training, 1,000 for development and 543 for test) where each phrase (as produced by the SMT decoder) has been annotated (as a phrase) by humans with four labels (see Section 7). This subset was selected after post-editing by filtering out translations with HTER=0 and with a HTER=0.30 and above, and then randomly selecting a subset large enough while fitting the annotation budget. The latter criterion was used to rule out sentences with too many errors, since these are generally too hard or impossible to annotate for errors by humans.

We used BRAT[7] to perform the phrase labelling. The annotator – a professional translator – was given the translations to annotate, along with their respective source sentence. We provided them with a preset environment where all translations were pre-labelled at phrase-level beforehand as OK. The annotator's task was then to change the labels of the incorrect phrases. The labelling was done following a 'pessimistic' approach, where we requested the annotator to only consider a phrase to be OK if all its words were OK. This task has two variants, as we describe later: Task3a, where a phrase annotation is propagate to all of its words and the task is framed as a word-level prediction task; and Task3b, where prediction is done at the phrase level. Table 3 shows the statistics of the resulting datasets for these variants of the task.

Since the data used for this task is a subset of the dataset of that used for Task 1, we selected as test sentences also a subset of the test set for Task 1.

### 4.3 Task 4

The document-level task data consists of short **product descriptions** translated from English to French, extracted from the Amazon Product Re-

[6] http://www.qt21.eu/

[7] https://brat.nlplab.org/

| | Train. | | Dev. | | Test | |
|---|---|---|---|---|---|---|
| Language pair | # Sentences | # Words | # Sentences | # Words | # Sentences | # Words |
| DE-EN | 25,963 | 493,010 | 1,000 | 18,817 | 1,254 | 23,522 |
| EN-DE-SMT | 26,273 | 442,074 | 1,000 | 16,565 | 1,926 | 32,151 |
| EN-DE-NMT | 13,442 | 234,725 | 1,000 | 17,669 | 1,023 | 17,649 |
| EN-LV-SMT | 11,251 | 225,347 | 1,000 | 20,588 | 1,315 | 26,661 |
| EN-LV-NMT | 12,936 | 258,125 | 1,000 | 19,791 | 1,448 | 28,945 |
| EN-CS | 40,254 | 728,815 | 1,000 | 18,315 | 1,920 | 34,606 |

Table 2: Statistics of the datasets used for Tasks 1 and 2: Total number of (source) sentences and words (after tokenisation) for training, development and test for each language pair and MT system type.

| **Task3a** | # Sentences | # Words | # BAD |
|---|---|---|---|
| Train. | 5,921 | 126,508 | 35,532 |
| Dev. | 1,000 | 28,710 | 6,153 |
| Test | 543 | 7,464 | 3,089 |
| **Task3b** | # Sentences | # Phrases | # BAD |
| Train. | 5,921 | 50,834 | 10,451 |
| Dev. | 1,000 | 8,566 | 1,795 |
| Test | 543 | 4,391 | 868 |

Table 3: Statistics of the data used for Task 3. Number of sentences, phrases, words and BAD labels for training, development and test.

| | # Documents | # Sentences | # Words |
|---|---|---|---|
| Train. | 1,000 | 6,003 | 129,099 |
| Dev. | 200 | 1,301 | 28,071 |
| Test | 269 | 1,652 | 39,049 |

Table 4: Statistics of the data used for Task 4. Number of documents, sentences and (target) words for training, development and test.

views dataset (McAuley et al., 2015; He and McAuley, 2016).[8] More specifically, the data is a selection of Sports and Outdoors product titles and descriptions in English which has been machine translated into French using a state of the art online neural MT system. The most popular products (those with more reviews) were chosen. This data poses interesting challenges for machine translation: titles and descriptions are often short and not always a complete sentence. Spans covering one or more tokens were annotated with error labels following fine-grained error taxonomy, as described in more detail in Section 8. The dataset statistics are presented in Table 4. This is the largest ever released collection with word-level errors manually annotated.

## 5 Task 1: Predicting sentence-level quality

This task consists in scoring (and ranking) translation sentences according to the proportion of their words that need to be fixed. HTER is used as quality score, i.e. the minimum edit distance between the machine translation and its manually post-edited version.

**Labels** Three labels were available: percentage of edits need to be fixed (HTER) (primary label), post-editing time in seconds, and counts of various types of keystrokes. The PET tool (Aziz et al., 2012)[9] was used to collect various types of information during post-editing. HTER labels were computed using the TERCOM tool[10] with default settings (tokenised, case insensitive, exact matching only), with scores capped to 1.

**Evaluation** Evaluation was performed against the true HTER label and/or ranking, using the following metrics:

- Scoring: Pearson's $r$ correlation score (primary metric, official score for ranking system submissions), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).
- Ranking: Spearman's $\rho$ rank correlation.

Statistical significance on Pearson $r$ was computed using the William's test.[11]

**Results** For Task 1, Tables 5, 6, 7 and 8 summarise the results for English–German, German–English, English–Latvian and English-Czech, respectively, ranking participating systems best to worst using Pearson's $r$ correlation as primary key.

[8] http://jmcauley.ucsd.edu/data/amazon/.

[9] https://github.com/ghpaetzold/PET

[10] https://github.com/jhclark/tercom

[11] https://github.com/ygraham/mt-qe-eval

Spearman's $\rho$ correlation scores should be used to rank systems for the ranking variant of the evaluation.

The top two systems for this task, the QEBrain model and UNQE models, show a large performance gap with respect to the rest of the systems, for both SMT and NMT data. It is interesting to note that both systems outperform the SHEF-PT system by a large margin. SHEF-PT is a reimplementation of the POSTECH system, which showed the top performance in 2017.

## 6 Task 2: Predicting word-level quality

This task evaluates the extent to which we can detect word-level errors in MT output. Often the overall quality of a translated segment is significantly harmed by specific errors in a small number of words. As in previous years, each token of the target sentence is labeled as OK/BAD based on an available post-edited sentence. In addition to this, this year we also took into consideration word omission errors and the detection of words in the source related to target side errors. These types of errors become particularly relevant in the context of NMT systems. The code to produce this new set of tags from any prior WMT corpora is available for download.[12]

**Target word labels** As in previous years, the binary labels for each target token (OK and BAD) were derived automatically by aligning each machine translated sentence with its post-edited counterpart sentence. The alignment at token-level was performed using the TERCOM tool. Default settings were used and shifts were disabled. Target tokens originating from insertion or substitution errors were labeled as BAD. All other tokens were labeled as OK.

**Gap and source word labels** To annotate deletion errors, gap 'tokens' between each word and at the beginning of each target sentence were introduced. These gaps tokens were labeled as BAD in the presence of one or more deletion errors and OK otherwise. To annotate the source words related to insertion or substitution errors in the machine translated sentence, the IBM Model 2 alignments from fastalign (Dyer et al., 2013) were used. Each token in the source sentence was aligned to the post-edited sentence. For each token in the post-edited sentence deleted or substituted in the machine translated text, the corresponding aligned source tokens were labeled as BAD. In this way, deletion errors also result in BAD tokens in the source, related to the missing words. All other words were labeled as OK.

**Evaluation** Analogously to last year's task, the primary evaluation metric is the multiplication of $F_1$-scores for the OK and BAD classes, denoted as $\mathrm{F_1}$-Mult. The same metric was applied to gap and source token labels. We also report $F_1$-scores for individual classes for completeness. We test the significance of the results using randomisation tests (Yeh, 2000) with Bonferroni correction (Abdi, 2007).

**Results** The results for Task 2 are summarised in Tables 9, 10, 11 and 12, ordered by the $\mathrm{F_1}$-mult metric.

The number of submissions per language pair was different, which limits any conclusions that can be made with respect to general rankings of systems. The English-German and German-English tasks – Tables 9, 10 – had the most systems participating. As in previous years, results in Task1 and Task2 are correlated. In this case the same system, QEBrain, wins both tasks for these language pairs. Since some of the other systems for Task 1 where specific for sentence-level prediction, the next system in the ranking is SHEF-PT, which lags behind by a margin slightly smaller than in Task 1. Another interesting result for this year is the differences between SMT and NMT datasets. For English-German, there is a clear drop in performance from SMT to NMT. This can be due to changes in the type of errors, or size of training sets, as we discuss in Section 9.

Regarding the novel task variants of detection of gaps and source words that lead to errors, only a few teams submitted systems. The performance for these tasks is lower, but correlated with the performance of the main word-level task – prediction of target word errors. It is worth noting that the QEBrain system obtains notable performance for gap error detection, almost doubling the performance of other (few) participating systems for SMT data.

The English-Latvian and English-Czech tasks had a lower number of participants, potentially due to the lower number of resources to pre-process data and pre-train models. It is interesting to note

[12] https://github.com/Unbabel/word-level-qe-corpus-builder

| Model | **Pearson** $r$ | MAE | RMSE | Spearman $\rho$ |
|---|---|---|---|---|
| SMT DATASET | | | | |
| • QEBrain DoubleBi w/ BPE+word-tok (ensemble) | 0.74 | 0.09 | 0.14 | 0.75 |
| QEBrain DoubleBi w/ BPE-tok | 0.73 | 0.10 | 0.14 | 0.75 |
| UNQE | 0.70 | 0.10 | 0.14 | 0.72 |
| TSKQE2 | 0.49 | 0.13 | 0.17 | 0.00 |
| SHEF-PT | 0.49 | 0.13 | 0.17 | 0.51 |
| TSKQE1 | 0.48 | 0.13 | 0.17 | 0.00 |
| UTartu/QuEst+Attention | 0.43 | 0.14 | 0.17 | 0.42 |
| UTartu/QuEst+Att+CrEmb3 | 0.42 | 0.14 | 0.17 | 0.42 |
| sMQE | 0.40 | 0.19 | 0.22 | 0.40 |
| RTM_MIX7 | 0.39 | 0.14 | 0.18 | 0.40 |
| RTM_MIX6 | 0.39 | 0.14 | 0.18 | 0.40 |
| SHEF-bRNN | 0.37 | 0.14 | 0.18 | 0.38 |
| BASELINE | 0.37 | 0.14 | 0.18 | 0.38 |
| uMQE | – | – | – | 0.38 |
| UAlacant** | 0.39 | 0.18 | 0.23 | 0.39 |
| NMT DATASET | | | | |
| • UNQE | 0.51 | 0.11 | 0.17 | 0.61 |
| • QEBrain DoubleBi w/ BPE+word-tok (ensemble) | 0.50 | 0.11 | 0.17 | 0.60 |
| • QEBrain DoubleBi w/ word-tok | 0.50 | 0.11 | 0.17 | 0.60 |
| TSKQE1 | 0.42 | 0.14 | 0.18 | 0.00 |
| TSKQE2 | 0.41 | 0.14 | 0.18 | 0.00 |
| SHEF-bRNN | 0.38 | 0.13 | 0.18 | 0.48 |
| SHEF-PT | 0.38 | 0.13 | 0.18 | 0.47 |
| UTartu/QuEst+Attention | 0.37 | 0.13 | 0.18 | 0.44 |
| sMQE | 0.37 | 0.21 | 0.24 | 0.44 |
| UTartu/QuEst+Att+CrEmb3 | 0.37 | 0.13 | 0.18 | 0.44 |
| BASELINE | 0.29 | 0.13 | 0.19 | 0.42 |
| uMQE | – | – | – | 0.40 |
| UAlacant** | 0.23 | 0.21 | 0.26 | 0.24 |
| RTM_MIX5** | 0.47 | 0.12 | 0.17 | 0.55 |

Table 5: Official results of the WMT18 Quality Estimation Task 1 for the **English–German** dataset. The winning submission is indicated by a •. Baseline systems are highlighted in grey, and ** indicates late submissions that were not considered for the official ranking of participating systems.

| Model | **Pearson** $r$ | MAE | RMSE | Spearman $\rho$ |
|---|---|---|---|---|
| • UNQE | 0.77 | 0.09 | 0.13 | 0.73 |
| • QEBrain DoubleBi w/ BPE+word-tok (ensemble) | 0.76 | 0.10 | 0.13 | 0.73 |
| • QEBrain DoubleBi w/ word-tok | 0.75 | 0.10 | 0.14 | 0.72 |
| sMQE | 0.65 | 0.12 | 0.15 | 0.60 |
| UTartu/QuEst+Att+CrEmb3 | 0.57 | 0.14 | 0.18 | 0.47 |
| SHEF-PT | 0.55 | 0.13 | 0.17 | 0.50 |
| UTartu/QuEst+Attention | 0.55 | 0.14 | 0.17 | 0.47 |
| SHEF-bRNN | 0.48 | 0.14 | 0.19 | 0.44 |
| BASELINE | 0.33 | 0.15 | 0.19 | 0.32 |
| UAlacant** | 0.63 | 0.12 | 0.17 | 0.60 |
| RTM_MIX5** | 0.54 | 0.13 | 0.17 | 0.49 |

Table 6: Official results of the WMT18 Quality Estimation Task 1 for the **German–English** dataset. The winning submission is indicated by a •. Baseline systems are highlighted in grey, and ** indicates late submissions that were not considered for the official ranking of participating systems.

| Model | **Pearson** $r$ | MAE | RMSE | Spearman $\rho$ |
|---|---|---|---|---|
| | SMT DATASET | | | |
| • UNQE | 0.62 | 0.12 | 0.16 | 0.58 |
| sMQE | 0.46 | 0.13 | 0.18 | 0.41 |
| UTartu/QuEst+Att+CrEmb3 | 0.40 | 0.16 | 0.20 | 0.32 |
| UTartu/QuEst+Attention | 0.40 | 0.15 | 0.19 | 0.32 |
| SHEF-bRNN | 0.40 | 0.14 | 0.19 | 0.33 |
| SHEF-PT | 0.38 | 0.14 | 0.19 | 0.33 |
| BASELINE | 0.35 | 0.16 | 0.19 | 0.35 |
| uMQE | – | – | – | 0.40 |
| UAlacant** | 0.36 | 0.20 | 0.26 | 0.34 |
| RTM_MIX** | 0.35 | 0.14 | 0.19 | 0.28 |
| | NMT DATASET | | | |
| • UNQE | 0.68 | 0.13 | 0.17 | 0.67 |
| sMQE | 0.58 | 0.15 | 0.19 | 0.57 |
| UTartu/QuEst+Att+CrEmb3 | 0.54 | 0.16 | 0.20 | 0.50 |
| UTartu/QuEst+Attention | 0.53 | 0.16 | 0.20 | 0.49 |
| SHEF-PT | 0.46 | 0.17 | 0.22 | 0.45 |
| BASELINE | 0.44 | 0.16 | 0.22 | 0.46 |
| SHEF-bRNN | 0.42 | 0.17 | 0.22 | 0.41 |
| uMQE | – | – | – | 0.54 |
| UAlacant** | 0.56 | 0.17 | 0.22 | 0.55 |
| RTM_MIX** | 0.54 | 0.16 | 0.20 | 0.50 |

Table 7: Official results of the WMT18 Quality Estimation Task 1 for the **English–Latvian** dataset. The winning submission is indicated by a •. Baseline systems are highlighted in grey, and ** indicates late submissions that were not considered for the official ranking of participating systems.

| Model | **Pearson** $r$ | MAE | RMSE | Spearman $\rho$ |
|---|---|---|---|---|
| • UNQE | 0.69 | 0.12 | 0.17 | 0.71 |
| SHEF-PT | 0.53 | 0.15 | 0.19 | 0.54 |
| SHEF-bRNN | 0.50 | 0.16 | 0.20 | 0.51 |
| UTartu/QuEst+Attention | 0.45 | 0.16 | 0.20 | 0.46 |
| UTartu/QuEst+Att+CrEmb3 | 0.41 | 0.17 | 0.21 | 0.40 |
| BASELINE | 0.39 | 0.17 | 0.21 | 0.41 |
| sMQE | 0.39 | 0.16 | 0.21 | 0.42 |
| uMQE | – | – | – | 0.42 |
| UAlacant** | 0.44 | 0.18 | 0.23 | 0.46 |
| RTM_MIX** | 0.52 | 0.15 | 0.20 | 0.53 |

Table 8: Official results of the WMT18 Quality Estimation Task 1 for the **English–Czech** dataset. The winning submission is indicated by a •. Baseline systems are highlighted in grey, and ** indicates late submissions that were not considered for the official ranking of participating systems.

the general low performance of systems on the English-Latvian NMT data: all systems are tied with the baseline in terms of F$_1$-mult. The reimplementation of the POSTECH system shows poor results on the NMT dataset, in this case it is unable to outperform the baseline. Results for English-Czech are very similar across systems.

## 7 Task 3: Predicting phrase-level quality

This level of granularity was first introduced in the shared task at WMT16. The goal is to predict MT quality at the level of phrases. In the 2016 edition, the data annotation was done automatically based on post-edits, as in Task 2, but this year humans directly labelled each phrase in context.

| SMT DATASET | Words in MT | | | GAPs in MT | | | Words in SRC | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** |
| ● QEBrain DoubleBi w/ BPE+word-tok (ensemble) | 0.68 | 0.92 | 0.62 | – | – | – | – | – | – |
| QEBrain DoubleBi w/ word-tok | 0.66 | 0.92 | 0.61 | 0.51 | 0.98 | 0.50 | – | – | – |
| SHEF-PT | 0.51 | 0.85 | 0.43 | 0.29 | 0.96 | 0.28 | 0.42 | 0.80 | 0.34 |
| CMU-LTI | 0.48 | 0.82 | 0.39 | – | – | – | – | – | – |
| SHEF-bRNN | 0.45 | 0.81 | 0.37 | 0.27 | 0.96 | 0.26 | 0.41 | 0.82 | 0.34 |
| BASELINE | 0.41 | 0.88 | 0.36 | – | – | – | – | – | – |
| Doc2Vec | 0.29 | 0.75 | 0.22 | – | – | – | – | – | – |
| BagOfWords | 0.28 | 0.73 | 0.20 | – | – | – | – | – | – |
| UAlacant** | 0.35 | 0.81 | 0.29 | 0.33 | 0.96 | 0.32 | – | – | – |
| RTM** | 0.33 | 0.88 | 0.29 | 0.26 | 0.98 | 0.25 | 0.17 | 0.86 | 0.14 |

| NMT DATASET | Words in MT | | | GAPs in MT | | | Words in SRC | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** |
| ● QEBrain DoubleBi w/ word-tok (using voting) | 0.48 | 0.91 | 0.44 | – | – | – | – | – | – |
| ● QEBrain DoubleBi w/ word-tok | 0.48 | 0.92 | 0.43 | – | – | – | – | – | – |
| CMU-LTI | 0.36 | 0.85 | 0.30 | – | – | – | – | – | – |
| SHEF-bRNN | 0.35 | 0.86 | 0.30 | 0.12 | 0.98 | 0.12 | 0.33 | 0.87 | 0.29 |
| SHEF-PT | 0.34 | 0.87 | 0.29 | 0.11 | 0.98 | 0.11 | 0.31 | 0.84 | 0.26 |
| BASELINE | 0.20 | 0.92 | 0.18 | – | – | – | – | – | – |
| UAlacant** | 0.23 | 0.86 | 0.19 | 0.12 | 0.98 | 0.12 | – | – | – |
| RTM** | 0.14 | 0.99 | 0.13 | 0.14 | 0.99 | 0.13 | 0.03 | 0.92 | 0.03 |

Table 9: Official results of the WMT18 Quality Estimation Task 2 for the **English–German** dataset. The winning submission is indicated by a ●. Baseline systems are in grey, and ** indicates late submissions that were not considered for the official ranking of participating systems.

|  | Words in MT |  |  | GAPs in MT |  |  | Words in SRC |  |  |
|---|---|---|---|---|---|---|---|---|---|
| Model | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** |
| • QEBrain DoubleBi w/ BPE+word-tok (ensemble) | 0.65 | 0.92 | 0.60 | – | – | – | – | – | – |
| • QEBrain DoubleBi w/ word-tok | 0.65 | 0.92 | 0.59 | – | – | – | – | – | – |
| BASELINE | 0.49 | 0.90 | 0.44 | – | – | – | – | – | – |
| SHEF-PT | 0.49 | 0.87 | 0.42 | 0.21 | 0.97 | 0.20 | 0.39 | 0.89 | 0.35 |
| CMU-LTI | 0.49 | 0.85 | 0.42 | – | – | – | – | – | – |
| SHEF-brNN | 0.45 | 0.87 | 0.39 | 0.20 | 0.97 | 0.19 | 0.37 | 0.87 | 0.32 |
| UAlacant** | 0.43 | 0.87 | 0.37 | 0.33 | 0.97 | 0.32 | – | – | – |
| RTM** | 0.38 | 0.90 | 0.34 | 0.15 | 0.98 | 0.14 | 0.12 | 0.90 | 0.11 |

Table 10: Official results of the WMT18 Quality Estimation Task 2 for the **German–English** dataset. The winning submission is indicated by a •. Baseline systems are in grey, and ** indicates late submissions that were not considered for the official ranking of participating systems.

|  | Words in MT |  |  | GAPs in MT |  |  | Words in SRC |  |  |
|---|---|---|---|---|---|---|---|---|---|
| Model | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** |
| • CMU-LTI | 0.56 | 0.80 | 0.45 | – | – | – | – | – | – |
| BASELINE | 0.53 | 0.83 | 0.44 | – | – | – | – | – | – |
| • SHEF-PT | 0.56 | 0.80 | 0.44 | 0.17 | 0.98 | 0.17 | 0.49 | 0.80 | 0.39 |
| SHEF-bRNN | 0.55 | 0.79 | 0.44 | 0.18 | 0.97 | 0.17 | 0.49 | 0.81 | 0.40 |
| UAlacant** | 0.42 | 0.75 | 0.32 | 0.15 | 0.95 | 0.15 | – | – | – |
| RTM** | 0.53 | 0.83 | 0.44 | 0.11 | 0.98 | 0.10 | 0.32 | 0.80 | 0.26 |

Table 11: Official results of the WMT18 Quality Estimation Task 2 for the **English–Czech** dataset. The winning submission is indicated by a •. Baseline systems are in grey, and ** indicates late submissions that were not considered for the official ranking of participating systems.

| SMT DATASET | Words in MT | | | GAPs in MT | | | Words in SRC | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** |
| • SHEF-PT | 0.42 | 0.87 | 0.36 | 0.14 | 0.97 | 0.14 | 0.35 | 0.86 | 0.30 |
| • SHEF-bRNN | 0.41 | 0.86 | 0.35 | 0.12 | 0.98 | 0.11 | 0.36 | 0.86 | 0.31 |
| BASELINE | 0.38 | 0.91 | 0.34 | – | – | – | – | – | – |
| CMU-LTI | 0.22 | 0.85 | 0.19 | – | – | – | – | – | – |
| UAlacant** | 0.27 | 0.82 | 0.22 | 0.11 | 0.96 | 0.11 | – | – | – |
| RTM** | 0.37 | 0.90 | 0.33 | 0.13 | 0.99 | 0.13 | 0.12 | 0.89 | 0.11 |

| NMT DATASET | Words in MT | | | GAPs in MT | | | Words in SRC | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** |
| • CMU-LTI | 0.52 | 0.83 | 0.43 | – | – | – | – | – | – |
| BASELINE | 0.49 | 0.86 | 0.42 | – | – | – | – | – | – |
| • SHEF-PT | 0.52 | 0.81 | 0.42 | 0.13 | 0.97 | 0.13 | 0.44 | 0.81 | 0.36 |
| • SHEF-bRNN | 0.50 | 0.83 | 0.42 | 0.12 | 0.94 | 0.11 | 0.44 | 0.80 | 0.36 |
| UAlacant** | 0.45 | 0.80 | 0.36 | 0.17 | 0.95 | 0.16 | – | – | – |
| RTM** | 0.43 | 0.85 | 0.37 | 0.08 | 0.98 | 0.08 | 0.20 | 0.84 | 0.17 |

Table 12: Official results of the WMT18 Quality Estimation Task 2 for the **English–Latvian** dataset. The winning submission is indicated by a •. Baseline systems are in grey, and ** late submissions that were not considered for the official ranking of participating systems.

**Labels** We used the phrase segmentation produced by the SMT decoder which generated the translations for the dataset. The phrases were annotated for errors using four classes: 'OK', 'BAD' – the phrase contain one or more errors, 'BAD_word_order' – the phrase is in an incorrect position in the sentence, and 'BAD_omission' – a word is missing before/after a phrase. This task in further subdivided in two subtasks: word-level prediction (Task3a), and phrase-level prediction (Task3b).

The data for Task3a propagates the annotation of each phrase to its words, and thus uses word-level segmentation for both source and machine-translated sentences, such that the task can be addressed as a word-level prediction task. In other words, all tokens in the target sentence are labelled according to the label of the phrase they belong to. Therefore, if the phrase is annotated as either 'OK', 'BAD' or 'BAD_word_order', all tokens (and gap tokens) within that phrase are labelled as either 'OK', 'BAD' or 'BAD_word_order'. To annotate omission errors, a gap token is inserted after each token and at the start of the sentence.

The data for Task3b has phrase-level segmentation with the labels assigned by the human annotator to each phrase. A gap token is inserted after each phrase and at the start of the sentence. The gap is labelled as follows: 'OK' or 'BAD_omission', where the latter indicates that one or more words are missing.

**Evaluation** Similarly to Task 2, our primary metric for predictions at word-level (Task3a) is the multiplication of the $F_1$ scores of the OK and BAD classes, $F_1$-Mult, while for predictions at phrase-level (Task3b), our primary metric is the phrase-level version of $F_1$-Mult. The same metrics were applied to gap and source token labels for both sub-tasks, along with $F_1$ scores for individual classes for completeness. We also report $F_1$ score for BAD_word_order labels on the target tokens for Task3b. We computed statistical significance of the results using randomised test with Bonferroni correction, as in Task 2.

**Results** The results of the phrase-level task are given in Tables 13 (Task3a) and 14 (Task3b), ordered by the $F_1$-Mult metric.

Comparing the results for Task3a with the results on German-English for Task 2 (Table 10), it can be observed a general degradation of the $F_1$ score on the BAD class, including for the baseline system. We attribute this phenomenon to the way the data for this task was created: for Task 2, the token labels were produced from post-editing, where each word was labelled independently from each others; while for this task, the token labels are deduced from a labelling at more coarse level (phrase), i.e. where words were not considered as individual tokens. Consequently, words that would be considered as correct during post-editing are here labelled as BAD, like to BAD phrase they belong to. The only two official submissions to this subtask (SHEF-PT and SHEF-bRNN) slightly outperform the baseline system, nevertheless without a statistically significant difference.

For the phrase-level predictions, the baseline system remains ahead by a significant margin of the only two official submissions, both from the University of Sheffield (SHEF-ATT-SUM and SHEF-PT). The overall performance in predicting phrases that are in incorrect position in a sentence (i.e. BAD_word_order) shows that this problem remains a very challenging task, as none of the submissions were able to obtain competitive $F_1$ score.

## 8 Task 4: Predicting document-level QE

This task consists in estimating a document-level quality score according to the amount of minor, major, and critical errors present in the translation. The predictions are compared to a ground-truth obtained from annotations produced by crowd-sourced human translators from Unbabel.[13]

**Labels** The data was annotated for errors at the word level using a fine-grained error taxonomy – Multidimensional Quality Metrics (MQM) (Lommel et al., 2014) – similar to the one used in (Sanchez-Torron and Koehn, 2016). MQM is composed of three major branches: accuracy (the translation does not accurately reflect the source text), fluency (the translation affects the reading of the text) and style (the translation has stylistic problems, like the use of a wrong register). These branches include more specific issues lower in the hierarchy. Besides the identification of an error and its classification according to this typology (by applying a specific tag), the errors receive a severity scale that reflects the impact of each error on the overall meaning, style, and fluency of the translation. An error can be *minor* (if it does

[13] http://www.unbabel.com.

| | **Words in MT** | | | **GAPs in MT** | | | **Words in SRC** | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** |
| ● SHEF-PT | 0.33 | 0.83 | 0.28 | 0.27 | 0.88 | 0.24 | 0.50 | 0.81 | 0.41 |
| ● SHEF-bRNN | 0.33 | 0.82 | 0.27 | 0.26 | 0.88 | 0.23 | 0.49 | 0.79 | 0.39 |
| BASELINE | 0.27 | 0.91 | 0.25 | – | – | – | – | – | – |
| RTM** | 0.16 | 0.90 | 0.15 | 0.10 | 0.94 | 0.10 | 0.10 | 0.84 | 0.08 |

Table 13: Official results of the WMT18 Quality Estimation Task 3a (word-level) for the **German–English** dataset. The winning submission is indicated by a ●. Baseline system is in grey, and ** indicates late submissions that were not considered for the official ranking of participating systems.

| | **Words in MT** | | | | **GAPs in MT** | | | **Words in SRC** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Model | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD_w_order | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** | $F_1$-BAD | $F_1$-OK | **$F_1$-mult** |
| BASELINE | 0.39 | 0.92 | 0.36 | 0.02 | – | – | – | – | – | – |
| ● SHEF-ATT-SUM | 0.29 | 0.76 | 0.22 | 0.11 | 0.10 | 0.94 | 0.10 | – | – | – |
| SHEF-PT | 0.23 | 0.81 | 0.18 | 0.08 | 0.11 | 0.93 | 0.10 | – | – | – |
| RTM** | 0.27 | 0.92 | 0.24 | 0.04 | 0.05 | 0.98 | 0.05 | 0.10 | 0.90 | 0.09 |

Table 14: Official results of the WMT18 Quality Estimation Task 3b (phrase-level) for the **German–English** dataset. The winning submission is indicated by a ●. Baseline system is in grey, and ** indicates late submissions that were not considered for the official ranking of participating systems.

not lead to a loss of meaning and it doesn't confuse or mislead the user), *major* (if it changes the meaning) or *critical* (if it changes the meaning and carry any type of implication, or could be seen as offensive).

Document-level scores were then generated from the word-level errors and their severity using the method described in Sanchez-Torron and Koehn (2016, footnote 6). Namely, denoting by $n$ the number of words in the document, and by $n_{\min}$, $n_{\text{maj}}$, and $n_{\text{cri}}$ the number of annotated minor, major, and critical errors, the final quality scores were computed as:

$$\text{MQM Score} = 1 - \frac{n_{\min} + 5n_{\text{maj}} + 10n_{\text{cri}}}{n} \quad (1)$$

Note that MQM values can be negative if the total severity exceeds the number of words.

**Evaluation** Submissions are evaluated as in Task 1 (see Section 5), in terms of Pearson's correlation $r$ between the true and predicted document-level scores.

**Results** The results of the document-level task are shown in Table 15. Due to the different numeric range, only the Pearson correlation scores are comparable to those of Task1. Comparing with the results for Task 1, it can be observed that the baseline system already obtains very high correlation. The neural model SHEF-PT-indomain outperforms the baseline by a modest margin, compared to the results obtained in Task 1.

| Model | **Pearson** $r$ | MAE |
|---|---|---|
| • SHEF-PT-indomain | 0.53 | 0.56 |
| BASELINE | 0.51 | 0.56 |
| SHEF-mtl-bRNN | 0.47 | 0.56 |
| SHEF-mtl-PT-indomain** | 0.52 | 0.57 |
| RTM_MIX1** | 0.11 | 0.58 |

Table 15: Official results of the WMT18 Quality Estimation Task 4 for the **English–French** dataset. The winning submission is indicated by a •. Baseline system is in grey, and ** indicates late submissions that were not considered for the official ranking of participating systems.

## 9 Discussion

In what follows, we discuss the main findings of this year's shared task based on the goals we had previously identified for it.

**Performance of QE approaches on the output of neural MT systems.** As previously mentioned, some of the data used for Tasks 1 and 2 is translated by both an SMT and an NMT system: the English-German and English-Latvian data. In Task 1, for English-German, the numbers of translations in the QE training data from the two systems are very different (26, 273 for SMT and 13, 442 for NMT) and thus no direct comparison can be made. This shows that the NMT system was of much higher quality than the SMT one, producing many more sentences that led to HTER=0. Of the sentences that remained, the average NMT quality in the training data is still higher: HTER=0.154 versus 0.253 for SMT. From the results, the top systems do considerably better on the SMT data ($r$=0.74 for SMT vs $r$=0.51 for NMT translations). This difference is also noticeable for the baseline system ($r$=0.37 for SMT vs $r$=0.29 for NMT translations). This could however be because of the difference in number of samples and/or significant differences in distributions of HTER scores in the two datasets. It is worth pointing out that the winning submissions are the same for both SMT and NMT translation: QEBrain and UNQE. In fact, QE system are ranked very similarly for the two types of translation.

For English-Latvian, the number of NMT and SMT QE training sentences is similar (12, 936 for NMT and 11, 251 for SMT). Their average HTER scores is also more comparable: 0.278 for NMT and 0.215 for SMT. The difference in QE system performance for this language pair is not as marked, but the trend is inverted when compared to English-German: QE systems do better on the NMT data (the top systems, UNQE, achieves $r$=0.62 for SMT vs $r$=0.68 for NMT translations, while the baseline achieves $r$=0.44 for SMT vs $r$=0.35 for NMT translations), This could be because of the lower differences in the distribution of HTER scores in both sets. The ranking of QE systems is exactly the same for both SMT and NMT translations.

In both cases, it is important to note that even though the initial datasets contained exactly the same source sentences for SMT and NMT, the sentences in the two final versions of the datasets for each language are not all the same, i.e. some NMT sentences may have gotten filtered for having HTER=0 while their SMT counterparts did not, and vice-versa. The main finding is that QE models seem to be robust to different types of translation, since their rankings are the same across datasets.

For Task 2, the trend is similar: QE systems for English-German also perform better on SMT translations than on NMT translations ($F_1$-Mult=0.62 for SMT vs $F_1$-Mult=0.44 for NMT), and the inverse is observed for English-Latvian ($F_1$-Mult=0.36 for SMT vs $F_1$-Mult=0.43 for NMT). The ranking of QE systems for the two types of translations differs more than for Task 1, especially for English-Latvian.

Task 4 uses NMT output only and it is hard to make any conclusions about whether the performance of the systems is good enough because this is the first time this task is organised. Generally speaking, this task proved hard, with the baseline system performing as well or better than the other submissions.

**Predictability of missing words in the MT output.** Only a subset of the systems that participated in Task 2 submitted results for missing word detection. From the results obtained it seems clear that while this task is more difficult than target word error detection, high scores could be attained for the SMT data. Due to the small number of submitted systems, it is unclear whether or not gap detection is more difficult for NMT data.

**Predictability of source words that lead to errors in the MT output.** Only a small set of teams submitted predictions for source words. From the submitted results, it can be observed that prediction of source words related to errors is a harder problem than detecting errors in the target language. This may be due to the fact that there may be more ambiguity with regards to which words should be related to errors in the target. In other words, in some cases a source word in a given context leads to incorrect translations, while in other cases the same source word in the same context will not lead to errors.

**Effectiveness of manually assigned labels for phrases.** With only one official (and one late) submission to the phrase-level QE task this year, it is hard to conclude whether having manual labels makes the task harder (although the baseline system performs as well as in the last edition), or whether the reason lies in the design of the neural models, which may not be suitable for this task.

**Quality prediction for documents from errors annotated at word-level with added severity judgements.** Since this is a new task and not many systems were submitted. Results show however that it is possible to attain Pearson correlation scores that are comparable with those of sentence-level post-editing effort prediction. The performance gap between the neural model and the SVM baseline is smaller than in Task 1, which may be an indication for further potential gains using new deep learning architectures tailored for document-level.

**Utility of additional evidence** To investigate the utility of detailed information logged during post-editing, we offered to participants other sources of information: post-editing time, keystrokes, and actual edits. Surprisingly, no participating system requested these additional labels, and therefore this remains an open question.

## 10 Conclusions

This year's edition of the QE shared task was the largest ever organised in many respects: number of tasks, number of languages, variety of tasks (three granularity levels), types of annotation (derived from post-editing or manual, source or target), and number of samples annotated.

Over the years, we have attempted to find a balance between keeping the shared task as close as possible to previous editions – so as to make some form of comparison across years possible – and proposing new tasks and new interesting challenges – so as to keep up to date with new developments in the field, such as neural machine translations. We believe the current set of tasks covers a broad enough range of challenges that are far from solved, such as improving performance given smaller sets of instances, predicting source words that lead to errors, predicting gaps, use of additional evidence from post-editing, etc.

In order to allow for future benchmarking on a 'blind' basis without access to the gold standard labels, we have set up CodaLab competitions that will remain open after this shared task. Any team can register and submit any number of systems (limited to five submissions per day per task and language pair) and get immediate feedback through the official evaluation metrics, as well as comparison to top submissions from other teams (on the leaderboard). Each team's best submission per task and language pair will feature on the leaderboard. The submission pages for each task are as follows, where languages and task variants are frames as 'phases':

- Sentence level: `https://competitions.codalab.org/competitions/19316`
- Word level: `https://competitions.codalab.org/competitions/19306`
- Phrase level: `https://competitions.codalab.org/competitions/19308`
- Document level: `https://competitions.codalab.org/competitions/19309`

## Acknowledgments

The data and annotations collected for Tasks 1, 2 and 3 was supported by the EC H2020 QT21 project (grant agreement no. 645452). The shared task organisation was also supported by the QT21 project, national funds through Fundação para a Ciência e Tecnologia (FCT), with references UID/CEC/50021/2013 and UID/EEA/50008/2013, and by the European Research Council (ERC StG DeepSPIN 758969). We would also like to thank Julie Belião and the Unbabel Quality Team for coordinating the annotation of the dataset used in Task 4.

## References

Hervé Abdi. 2007. The bonferroni and šidák corrections for multiple comparisons. *Encyclopedia of measurement and statistics*, 3:103–107.

Wilker Aziz, Sheila Castilho Monteiro de Sousa, and Lucia Specia. 2012. Pet: a tool for post-editing and assessing machine translation. In *Eighth International Conference on Language Resources and Evaluation*, LREC, pages 3982–3987, Istanbul, Turkey.

Prasenjit Basu, Santanu Pal, and Sudip Kumar Naskar. 2018. Keep it or not: Word level quality estimation for post-editing. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Brussels, Belgium. Association for Computational Linguistics.

Ergun Biçici. 2017. Predicting translation performance with referential translation machines. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 540–544, Copenhagen, Denmark. Association for Computational Linguistics.

Ergun Biçici. 2018. Rtm results for predicting translation performance. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Brussels, Belgium. Association for Computational Linguistics.

Ergun Bicici. 2013. Referential translation machines for quality estimation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 343–351, Sofia, Bulgaria. Association for Computational Linguistics.

Ondrej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Eighth Workshop on Statistical Machine Translation*, WMT, pages 1–44, Sofia, Bulgaria.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Ninth Workshop on Statistical Machine Translation*, WMT, pages 12–58, Baltimore, Maryland.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012.

Findings of the 2012 workshop on statistical machine translation. In *Seventh Workshop on Statistical Machine Translation*, WMT, pages 10–51, Montréal, Canada.

Melania Duma and Wolfgang Menzel. 2018. The benefit of pseudo-reference translations in quality estimation of mt output. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Brussels, Belgium. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Thierry Etchegoyhen, Eva Martínez Garcia, and Andoni Azpeitia. 2018. Supervised and unsupervised minimalist quality estimators: Vicomtech's participation in the wmt 2018 quality estimation task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Brussels, Belgium. Association for Computational Linguistics.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517. International World Wide Web Conferences Steering Committee.

Junjie Hu, Wei-Cheng Chang, Yuexin Wu, and Graham Neubig. 2018. Contextual encoding for translation quality estimation. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Brussels, Belgium. Association for Computational Linguistics.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018a. DeepQuest: a framework for neural-based quality estimation. In *Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics: Technical Papers*, Santa Fe, New Mexico.

Julia Ive, Carolina Scarton, Frédéric Blain, and Lucia Specia. 2018b. Sheffield submissions for the wmt18 quality estimation shared task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Brussels, Belgium. Association for Computational Linguistics.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, pages 562–568, Copenhagen, Denmark. Association for Computational Linguistics.

Julia Kreutzer, Shigehiko Schamoni, and Stefan Riezler. 2015. QUality Estimation from ScraTCH (QUETCH): Deep Learning for Word-level Translation Quality Estimation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 297–303, Lisboa, Portugal. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196.

Maoxi Li, Qingyu Xiang, Zhiming Chen, and Mingweng Wang. 2018. A unified neural network for quality estimation of machine translation. *IEICE Trans. Information and Systems*, E101-D(9).

Varvara Logacheva, Chris Hokamp, and Lucia Specia. 2016. Marmot: A toolkit for translation quality estimation at the word level. In *Tenth International Conference on Language Resources and Evaluation*, LREC, pages 3671–3674, Portoroz, Slovenia.

Arle Richard Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2014. Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. *Tradumàtica: tecnologies de la traducció*, 0(12):455–463.

Ngoc Quang Luong, Laurent Besacier, and Benjamin Lecouteux. 2014. Word confidence estimation for smt n-best list re-ranking. In *Proceedings of the EACL 2014 Workshop on Humans and Computer-assisted Translation*, pages 1–9, Gothenburg, Sweden. Association for Computational Linguistics.

André F. T. Martins, Ramón Astudillo, Chris Hokamp, and Fabio Kepler. 2016. Unbabel's participation in the wmt16 word-level translation quality estimation shared task. In *Proceedings of the First Conference on Machine Translation*, pages 806–811, Berlin, Germany. Association for Computational Linguistics.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM.

Sylvain Raybaud, David Langlois, and Kamel Smaili. 2011. "this sentence is wrong." detecting errors in machine-translated sentences. *Machine Translation*, 25(1):1–34.

Felipe Sánchez-Martíínez, Miquel Esplà-Gomis, and Mikel L. Forcada. 2018. Ualacant machine translation quality estimation at wmt 2018: a simple approach using phrase tables and feed-forward neural networks. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Brussels, Belgium. Association for Computational Linguistics.

Marina Sanchez-Torron and Philipp Koehn. 2016. Machine translation quality and post-editor productivity. *AMTA 2016, Vol.*, page 16.

Lucia Specia, Kim Harris, Frédéric Blain, Aljoscha Burchardt, Viviven Macketanz, Inguna Skadina, Matteo Negri, , and Marco Turchi. 2017. Translation quality and productivity: A study on rich morphology languages. In *Machine Translation Summit XVI*, pages 55–71, Nagoya, Japan.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jiayi Wang, Kai Fan, Bo Li, Fengming Zhou, Boxing Chen, Yangbin Shi, and Luo Si. 2018. Alibaba submission for wmt18 quality estimation task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Brussels, Belgium. Association for Computational Linguistics.

Elizaveta Yankovskaya, Andre Tattar, and Mark Fishel. 2018. Quality estimation with force-decoded attention and cross-lingual embeddings. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Brussels, Belgium. Association for Computational Linguistics.

Alexander Yeh. 2000. More Accurate Tests for the Statistical Significance of Result Differences. In *Coling-2000: the 18th Conference on Computational Linguistics*, pages 947–953, Saarbrücken, Germany.

# Findings of the WMT 2018 Shared Task on Automatic Post-Editing

**Rajen Chatterjee**[(1)], **Matteo Negri**[(1)], **Raphael Rubino**[(2)], **Marco Turchi**[(1)]
[(1)] Fondazione Bruno Kessler, Trento, Italy
[(2)] Universität des Saarlandes & DFKI, Saarbrücken, Germany
{chatterjee,negri,turchi}@fbk.eu
{raphael.rubino}@dfki.de

## Abstract

We present the results from the fourth round of the WMT shared task on MT Automatic Post-Editing. The task consists in automatically correcting the output of a "black-box" machine translation system by learning from human corrections. Keeping the same general evaluation setting of the three previous rounds, this year we focused on one language pair (English-German) and on domain-specific data (Information Technology), with MT outputs produced by two different paradigms: phrase-based (PBSMT) and neural (NMT). Five teams submitted respectively 11 runs for the PBSMT subtask and 10 runs for the NMT subtask. In the former subtask, characterized by original translations of lower quality, top results achieved impressive improvements, up to -6.24 TER and +9.53 BLEU points over the baseline "*do-nothing*" system. The NMT subtask proved to be more challenging due to the higher quality of the original translations and the availability of less training data. In this case, top results show smaller improvements up to -0.38 TER and +0.8 BLEU points.

## 1 Introduction

The WMT shared task on MT Automatic Post-Editing (APE), this year at its fourth round, aims to evaluate systems for the automatic correction of errors in a machine-translated text. As pointed out by (Chatterjee et al., 2015), from the application point of view the task is motivated by its possible uses to:

- Improve MT output by exploiting information unavailable to the decoder, or by performing deeper text analysis that is too expensive at the decoding stage;
- Cope with systematic errors of an MT system whose decoding process is not accessible;
- Provide professional translators with improved MT output quality to reduce (human) post-editing effort;
- Adapt the output of a general-purpose MT system to the lexicon/style requested in a specific application domain.

The 2018 round of the task proposed participants with the same evaluation setting of the three previous editions (Bojar et al., 2015; Bojar et al., 2016; Bojar et al., 2017), in which the output of an unknown "black box" MT engine has to be automatically corrected by learning from human revisions of translations produced by the same engine.

This year, the task focused on one language pair[1] (English-German) and, in continuity with the 2016 and 2017 rounds, on data coming from the Information Technology domain. The main novelty was represented by the use of training/test data including, for the same source sentences, translations produced by two different MT technologies: phrase-based (in continuity with 2016 and 2017) and neural (for the first time). On one side, keeping language and domain unchanged was meant to measure the technology progress over the past. On the other side, extending the evaluation to NMT-derived data was meant to explore the effectiveness of APE techniques, which now migrated to the neural paradigm, to correct data obtained with the same paradigm.

In terms of participants and submitted runs, 5 teams produced respectively 11 runs for the PBSMT subtask and 10 runs for the NMT subtask.

[1] As opposed to the 2017 round, in which both English-German and German-English data were considered.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 710–725
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64079

All submissions were produced by neural APE systems. All the teams experimented with the Transformer architecture (Vaswani et al., 2017), either directly or by adapting it to the task (see Section 2.1). The two synthetic corpora provided as additional training material (see Section 2.1) were also extensively used.

In terms of results, on PBSMT data, the last year's trend is confirmed: the migration to the neural approach to APE yielded significant quality gains to the output of phrase-based MT systems. However, while in 2017 the largest improvements with respect to the baseline were respectively -4.9 TER and +7.6 BLEU, this year the distance is even larger: -6.24 TER and +9.53. On NMT data, the gains are less evident, with the largest improvements over the baseline of -0.38 TER and +0.8 BLEU.

The large difference in terms of quality gains yield by APE can be explained in several ways. One is the different amount of in-domain training data available: in the PBSMT subtask, they comprise 28,000 instances while, in the NMT subtask, they are less than 14,000.[2] Another reason is the different MT output quality in the two datasets. Indeed, TER and BLEU scores for the PBSMT test set are respectively 24.24 and 62.99 while, in the NMT test set, they reach considerably better values of 16.84 and 74.73. Altogether, these differences contributed to make the NMT subtask more challenging, participants' scores concentrated in small TER/BLEU ranges close to the baseline and the overall results harder to interpret.

## 2 Task description

Similar to previous years, participants were provided with training and development data consisting of (*source*, *target*, *human post-edit*) triplets, and were asked to return automatic post-edits for a test set of unseen (*source*, *target*) pairs.

### 2.1 Data

For this year's round, the APE task focused on one language pair, English-German, and on data coming from the `Information Technology` (IT) domain. As emerged from the previous evaluations, the selected target domain is specific and repetitive enough to allow supervised systems to learn from the training set useful correction patterns that are also re-applicable to the test set.

Training and development sets consist of (*source*, *target*, *human post-edit*) triplets in which:

- The source (SRC) is a tokenized English sentence with length between 3 and 30 tokens;
- The target (TGT) is a tokenized German translation of the source, which is produced by a black-box system unknown to participants. Translations were produced with two different technologies, so to obtain two different subtasks and evaluation scenarios. The first subtask, in continuity with the past, focused on handling translations produced by a domain-adapted phrase-based system (PBSMT subtask).[3] The second subtask (NMT subtask) focused on handling translations produced by a domain-adapted neural system.[4]
- The human post-edit (PE) is a manually-revised version of the target, which was produced by professional translators.

Test data consists of (*source*, *target*) pairs having similar characteristics of those in the training set. Human post-edits of the test target instances are left apart to measure system performance.

For the **PBSMT subtask**, the *training* data available include: *i)* all the 15,000 triplets (training, development and test) released for the 2016 round of the APE task and *ii)* the 13,000 training and test triplets released for the 2017 round, for a total of 28,000 instances. The *test* set consists of 2,000 newly-released instances.

For the **NMT subtask**, the *training* and development set respectively consist of 13,442 and 1,000 triplets, while the test set comprises 1,023 instances.

[2] In addition to these small in-domain training sets, which were released by the organizers over the years, participants were also provided with large synthetic corpora described in Section 2.1.

[3] We used a phrase-based MT system trained with generic and in-domain parallel training data, leveraging prereordering techniques (Herrmann et al., 2013), and taking advantage of POS and word class-based language models.

[4] The NMT system was trained with generic and in-domain parallel training data using the attentional encoder-decoder architecture (Bahdanau et al., 2014) implemented in the Nematus toolkit (Sennrich et al., 2017). We used byte-pair encoding (Sennrich et al., 2016) for vocabulary reduction, mini-batches of 100, word embeddings of 500 dimensions, and gated recurrent unit layers of 1,024 units. Optimization was done using Adam and by re-shuffling the training set at each epoch.

|  | Instances |  |  |  |
|---|---|---|---|---|
|  | Corpus | 2016 | 2017 | 2018 |
| PBSMT | Train | 12,000 | 11,000 | - |
|  | Dev | 1,000 | - | - |
|  | Test | 2,000 | 2,000 | 2,000 |
| NMT | Train | - | - | 13,442 |
|  | Dev | - | - | 1,000 |
|  | Test | - | - | 1,023 |
| Additional Resources | Artificial |  | 4.5M |  |
|  | eSCAPE-PBSMT | - | - | 7,258,533 |
|  | eSCAPE-NMT | - | - | 7,258,533 |

Table 1: Data statistics.

|  | APE15 | APE16 | APE17 | APE17 | APE18 | APE18 |
|---|---|---|---|---|---|---|
| Language | En-Es | En-De | En-De | De-En | En-De | En-De |
| Domain | News | IT | IT | Medical | IT | IT |
| MT type | PBSMT | PBSMT | PBSMT | PBSMT | PBSMT | NMT |
| Repetition Rate SRC | 2.905 | 6.616 | 7.216 | 5.225 | 7.139 | 7.111 |
| Repetition Rate TGT | 3.312 | 8.845 | 9.531 | 6.841 | 9.471 | 9.441 |
| Repetition Rate PE | 3.085 | 8.245 | 8.946 | 6.293 | 8.934 | 8.941 |
| TER (↓) | 23.84 | 24.76 | 24.48 | 15.55 | 24.24 | 16.84 |
| BLEU (↑) | n/a | 62.11 | 62.49 | 79.54 | 62.99 | 74.73 |

Table 2: Repetition Rate and translation quality (TER/BLEU of TGT) of the WMT15, WMT16, WMT17 and WMT18 APE task data. Grey columns refer to data covering different language pairs and domains with respect to this year's evaluation round.

Participants were also provided with additional training material for both the subtasks. One resource (called "Artificial" in Table 1) is the corpus of 4.5 million artificially-generated post-editing triplets used by the 2016 winning system (Junczys-Dowmunt and Grundkiewicz, 2016). This corpus was widely used by participants in the 2017 round of the APE task. The other resource is the English-German section of the eSCAPE corpus (Negri et al., 2018). It comprises 14.5 million instances, which were artificially generated both via phrase-based and neural translation (7.25 millions each) of the same source sentences.

Table 1 provides basic statistics about the data, which was released by the European Project QT21 (Specia et al., 2017).

In addition, Table 2 provides a view of the data from a task difficulty standpoint. For each dataset released in the four rounds of the APE task, we report the repetition rate of SRC, TGT and PE elements, as well as the TER (Snover et al., 2006) and the BLEU score (Papineni et al., 2002) of the TGT elements (i.e. the original target translations).

The repetition rate measures the repetitiveness inside a text by looking at the rate of non-singleton n-gram types (n=1...4) and combining them using the geometric mean. Larger values indicate a higher text repetitiveness and, as discussed in (Bojar et al., 2016; Bojar et al., 2017), suggest a higher chance of learning from the training set correction patterns that are applicable also to the test set. In the previous rounds of the task, we considered the large differences in repetitiveness across the datasets as a possible explanation for the variable gains over the baseline obtained by participants. In this perspective, the low system performance observed in the APE15 task and in the APE17 German-English subtask was in part ascribed to the low repetition rate in the data. In contrast, much higher repetition rates in the data likely contributed to facilitate the problem in the APE16 task and in the APE17 English-German subtask, in which most of the participants achieved significant gains over the baseline. For this year's data, values are in line with these two previous rounds.

The TER (↓) and BLEU (↑) scores reported in Table 2 are computed using the human post-edits as reference. As discussed in (Bojar et al., 2017), numeric evidence of a higher quality of the original translations can indicate a smaller room for improvement for APE systems (having, at the same time, less to learn during training and less to cor-

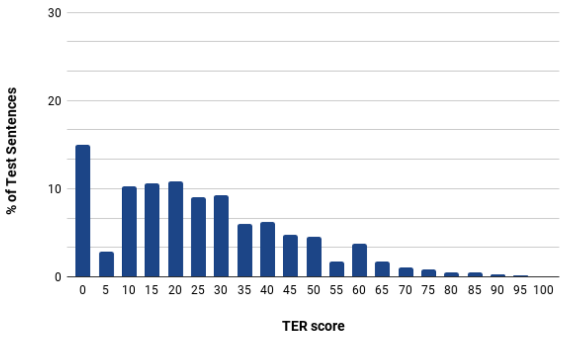


Figure 1: TER distribution in the **PBSMT** test set

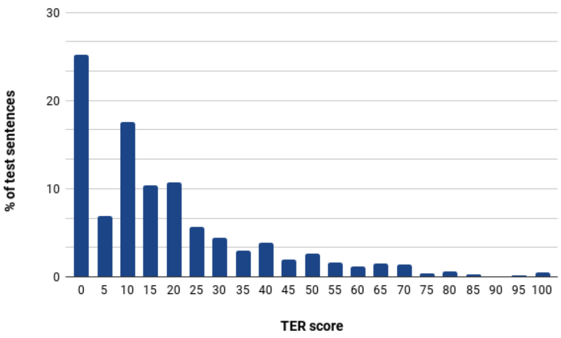


Figure 2: TER distribution in the **NMT** test set

rect at test stage). On one side, indeed, training on good (or near-perfect) automatic translations can drastically reduce the number of learned correction patterns. On the other side, testing on similarly good translations can drastically reduce the number of corrections required and the applicability of the learned patterns, thus making the task more difficult. As observed in the previous APE evaluation rounds, there is a noticeable correlation between translation quality and systems' performance. In 2016 and 2017, on English-German data featuring a similar level of quality (24.76/24.48 TER, 62.11/62.49 BLEU), the top neural systems achieved significant improvements over the baseline (-3.24 TER and +5.54 BLEU in 2016, -4.88 TER and +7.58 BLEU in 2017). In 2017, on higher quality German-English data (15.55 TER, 79.54 BLEU), the observed gains were much smaller (-0.26 TER, +0.28 BLEU). These numbers are not directly comparable since the higher quality 2017 data cover a different language pair and belong to a different domain. Nevertheless, as discussed in Section 4, this year's results confirm the correlation between the quality of the initial translations and the actual potential of APE.

Further indications about the difficulty of the two subtasks are provided by Figures 1 and 2, which plot the TER distribution for the items in the two test sets. As can be seen, the PBSMT test data are more distributed in terms of quality, with 50% of the items in the first five TER bins. Similar to last year, what makes a big difference between the two test sets is the proportion of "perfect" test instances having TER=0 (i.e. items that should not be modified by the APE systems). For the PBSMT subtask they are 15.0% of the total, a value similar to the APE17 English-German task in which participants achieved large baseline improvements. For the NMT subtask, they are 25.2% of the total: much less than the proportion of the challenging APE17 German-English data (45.0%) but still a considerably higher value compared to the PBSMT subtask. For these test items, any correction made by the APE systems will be treated as unnecessary and penalized by automatic evaluation metrics. This problem calls for conservative and precise systems able to properly fix errors only in the remaining test items, leaving the "perfect" ones unmodified.

## 2.2 Evaluation metrics

System performance was evaluated both by means of automatic metrics and manually. Automatic metrics were used to compute the distance between *automatic* and *human* post-edits of the machine-translated sentences present in the test sets (i.e. for each of the target sentences in the PBSMT and NMT test sets). To this aim, TER and BLEU (case-sensitive) were respectively used as primary and secondary evaluation metrics. Systems were ranked based on the average TER calculated on the test set by using the TERcom[5] software: lower average TER scores correspond to higher ranks. BLEU was computed using the multi-bleu.perl package[6] available in MOSES.

Manual evaluation was conducted via direct human assessment (Graham et al., 2016) performed by professional translators and proficient translation students, as discussed in Section 6.

## 2.3 Baseline

In continuity with the previous rounds, the official baseline results were the TER and BLEU scores

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

[6] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

| ID | Participating team |
|---|---|
| DFKI-MLT | German Research Center for Artificial Intelligence, Germany (Pylypenko and Rubino, 2018) |
| FBK | Fondazione Bruno Kessler, Italy (Tebbifakhr et al., 2018) |
| MS_UEdin | Microsoft, USA & University of Edinburgh, Poland (Junczys-Dowmunt and Grundkiewicz, 2018) |
| POSTECH | Pohang University of Science and Technology, South Korea (Shin and Lee, 2018) |
| USAAR_DFKI | Saarland University & German Research Center for Artificial Intelligence, Germany (Pal et al., 2018) |

Table 3: Participants in the WMT18 Automatic Post-Editing task.

calculated by comparing the raw MT output with the human post-edits. In practice, the baseline APE system is a "*do-nothing*" system that leaves all the test targets unmodified. Baseline results, the same shown in Table 2, are also reported in Tables 4 and 5 for comparison with participants' submissions.[7]

For each submitted run, the statistical significance of performance differences with respect to the baseline was calculated with the bootstrap test (Koehn, 2004).

## 3 Participants

Five participating teams submitted a total of 11 runs for the PBSMT subtask and 10 runs for the NMT subtask. Participants are listed in Table 3, and a short description of their systems is provided in the following.

**German Research Center for Artificial Intelligence - MLT group.** The DFKI-MLT's participation is based on a single APE model that is jointly trained to handle PBNMT and NMT outputs. This was achieved by adding, at the beginning of every MT segment to be corrected, a specific token indicating which type of MT system was used to produce it and from which training corpus the segment pair was extracted. (i.e. the WMT training data, the artificial training data presented in (Junczys-Dowmunt and Grundkiewicz, 2016), or the eSCAPE corpus (Negri et al., 2018)). The submitted runs were obtained with two neural architectures. One ("*LSTM*") is an attentional RNN with gated units based on (Bahdanau et al., 2014) and implemented in OpenNMT (Klein et al., 2017). The other is the multi-head attention-only network (Vaswani et al., 2017) implemented in the Marian NMT toolkit (Junczys-Dowmunt et al., 2018). For the attention-only approach, two models (i.e. "*Transf.base*" and "*Transf.large*") were trained with different configurations in terms of parallel attention layers (4 and 8 respectively).

**Fondazione Bruno Kessler.** FBK's system improves the multi-source neural approach adopted in (Chatterjee et al., 2017). The improvements target lower complexity of the architecture and, in turn, higher efficiency without loss in performance. To this aim, the proposed solution relies on the Transformer architecture (Vaswani et al., 2017), which was modified to incorporate multiple encoders, thereby leveraging information also from the source sentences. In addition, similar to (Hokamp, 2017), the system exploits minimum-risk training for fine-tuning (Shen et al., 2016) to avoid exposure bias and to be consistent with the automatic evaluation metrics used for the task. Finally, in order to reduce the vocabulary size, the system applies *ad hoc* pre-processing for the German language by re-implementing the pipeline developed by the best system at the WMT'17 Translation task (Huck et al., 2017). In addition to the data released for the task, training is performed by taking advantage of both the artificial data provided by (Junczys-Dowmunt and Grundkiewicz, 2016) and the eSCAPE corpus (Negri et al., 2018). The submitted runs, which rely on the same multi-source architecture and pre-processing step, differ in the loss function used, which is either minimum-risk training alone ("*MRT*"), or its linear combination with maximum likelihood estimation ("*MRT+MLE*").

**Microsoft & University of Edinburgh.** MS_UEdin's neural APE system is based on the dual-source Transformer models available in Marian (Junczys-Dowmunt et al., 2018). The models are trained with tied embeddings across all embeddings matrices and shared parameters for all the encoders. The dual-source Transformer model is implemented by stacking an additional target-source multi-head component on the previ-

[7] In addition to the *do-nothing* baseline, in previous rounds we also compared systems' performance with a re-implementation of the phrase-based approach firstly proposed by Simard et al. (2007), which represented the common backbone of APE systems before the spread of neural solutions. As shown in (Bojar et al., 2016; Bojar et al., 2017), the steady progress of neural APE technology has made the phrase-based solution not competitive with current methods reducing the importance of having it as an additional term of comparison.

ous multi-head component, one for each encoder. Each multi-head attention block is followed by a skip connection from the previous input and layer normalization. Each encoder corresponds exactly to the implementation from (Vaswani et al., 2017), but with common parameters. The decoder consists of a self-attention block, a target-to-source attention block, another target-to-source attention block and a feed-forward network. Apart from this modification, the system follows the transformer-base configuration from (Vaswani et al., 2017). The synthetic data provided by Junczys-Dowmunt and Grundkiewicz (2016) and the eSCAPE corpus (Negri et al., 2018) were both used during training, the latter being splitted into subsets by means of domain selection algorithms aimed to isolate useful portions for the APE target domain (IT). Final submissions were produced with an ensemble of models trained on the different subsets.

**Pohang University of Science and Technology.** POSTECH's system is a multi-encoder model that extends the Transformer implementation in the Tensor2tensor library in order to model the relation between the original translation produced by the MT system and the ideal translation produced by the human. System training was performed by taking advantage of the synthetic data released by Junczys-Dowmunt and Grundkiewicz (2016), which were divided into a smaller (526,368 instances) and a larger sub-portion (4,391,180) and used in a training process based on step-wise data reductions. The final submissions were obtained from the best single models (top-1), as well as their combination with different ensembling techniques ("*fix5*" – the top-5 models in a fixed checkpoint frequency and "*var5*" – five top-1 models for various checkpoint frequencies).

**Saarland University & German Research Center for Artificial Intelligence.** USAAR_DFKI's APE system extends the transformer-based NMT architecture by using two encoders, a joint encoder, and a single decoder. The presented model concatenates two separate self-attention-based encoders ($enc_{src}$ and $enc_{mt}$) and passes this sequence through another self-attended joint encoder ($enc_{src,mt}$) to ensure capturing dependencies between $src$ and $mt$. Finally, this joint encoder is fed to the decoder which follows a similar architecture as described in (Vaswani et al., 2017). A comparison between this multi-source architecture (i.e, $\{src, mt\} \rightarrow pe$), a monolingual transformer model (i.e., $mt \rightarrow pe$) and an ensemble of the multi-source $\{src, mt\} \rightarrow pe$ and single-source $mt \rightarrow pe$ models showed better results from the ensemble model (both in the PBSMT and the NMT subtasks), which was hence used for the final submission.

## 4 Results

Participants' results are shown in Tables 4 (PBSMT subtask) and 5 (NMT subtask). The submitted runs are ranked based on the average TER (case-sensitive) computed using human post-edits of the MT segments as reference, which is the APE task primary evaluation metric ("*TER (pe)*"). The two tables also report the BLEU score computed using human post-edits ("*BLEU (pe)*" column), which represents our secondary evaluation metric. These results are commented in Section 4.1.

The last four columns of both tables report the TER/BLEU scores computed using external references ("*TER (ref)*" and "*BLEU (ref)*") and the multi-reference TER/BLEU scores computed using human post-edits and external references ("*TER (pe+ref)*" and "*BLEU (pe+ref)*"). These results are commented in Section 4.2.

As a general remark about the two subtasks, we observe that in the NMT subtask, with all the metrics considered, the performance differences between the submitted runs are smaller (and more often not significant) compared to the PBSMT subtask. As discussed in the next sections, this makes it difficult to draw firm conclusions from the analysis of Table 5.

### 4.1 Automatic metrics computed using human post-edits

In terms of systems' ranking, the primary ("*TER (pe)*") and secondary evaluation metric ("*BLEU (pe)*") produce similar results.[8] On both the subtasks, the small differences in the TER-based and BLEU-based ranking concern a different ordering of the runs submitted by specific teams: one for the PBSMT subtask (in which FBK's primary submission is slightly better than the contrastive one in terms of BLEU) and two for the NMT subtask (in which POSTECH's and DFKI-MLT's best

[8] The correlation between the ranks obtained by the two metrics is 0.99 for the PBSMT subtask and 0.97 for the NMT subtask.

| ID | TER<br>(pe) | BLEU<br>(pe) | TER<br>(ref) | BLEU<br>(ref) | TER<br>(pe+ref) | BLEU<br>(pe+ref) |
|---|---|---|---|---|---|---|
| MS_UEdin Primary | 18.0 | 72.52 | 42.66 | 42.93 | 17.03 | 76.7 |
| FBK Contrastive (MRT+MLE) | 18.62 | 71.04 | 43.29 | 41.99 | 17.79 | 75.19 |
| FBK Primary (MRT) | 18.94 | 71.22 | 43.74 | 41.67 | 18.18 | 74.96 |
| POSTECH Contrastive (fix5) | 19.63 | 69.87 | 43.91 | 41.46 | 18.82 | 74.02 |
| POSTECH Primary | 19.72 | 69.8 | 43.95 | 41.45 | 18.9 | 73.94 |
| POSTECH Contrastive (var5) | 19.74 | 69.7 | 43.98 | 41.35 | 18.9 | 73.93 |
| USAAR_DFKI Primary | 22.69 | 66.16 | 46.08 | 39.26 | 21.98 | 69.73 |
| USAAR_DFKI* | 22.88 | 66.05 | 46.09 | 39.27 | 22.13 | 69.68 |
| DFKI-MLT Primary (Transf.large) | 24.19† | 63.4 | 47.98 | 36.81 | 23.68† | 66.66 |
| Baseline | 24.24 | 62.99 | 48.33 | 36.42 | 23.76 | 66.21 |
| DFKI-MLT Contrastive (Transf.base) | 24.5† | 62.78† | 48.27† | 36.61† | 24.04† | 66.11† |
| DFKI-MLT Contrastive (LSTM) | 25.3 | 62.1 | 48.55† | 36.19† | 24.74 | 65.33 |

Table 4: Results for the WMT18 APE **PBSMT subtask** – average TER (↓), BLEU score (↑). The symbol "†" indicates a difference from the MT baseline that is not statistically significant. The symbol "*" indicates a late submission by the USAAR_DFKI team.

| ID | TER<br>(pe) | BLEU<br>(pe) | TER<br>(ref) | BLEU<br>(ref) | TER<br>(pe+ref) | BLEU<br>(pe+ref) |
|---|---|---|---|---|---|---|
| FBK Primary (MRT) | 16.46 | 75.53 | 42.26† | 44.3† | 16.03 | 77.36 |
| MS_UEdin Primary | 16.5 | 75.44 | 42.15† | 44.46† | 16.05 | 77.49 |
| FBK Contrastive (MRT+MLE) | 16.55 | 75.38 | 42.15† | 44.37† | 16.09 | 77.28 |
| POSTECH Contrastive (top1) | 16.7† | 75.14 | 42.16† | 44.29† | 16.23 | 77.16 |
| POSTECH Primary (fix5) | 16.71† | 75.13 | 42.2† | 44.21† | 16.23 | 77.12 |
| POSTECH Contrastive (var5) | 16.71† | 75.2 | 42.19† | 44.27† | 16.23 | 77.15 |
| Baseline | 16.84 | 74.73 | 42.24 | 44.22 | 16.27 | 76.83 |
| USAAR_DFKI Primary | 17.23 | 74.22 | 42.51† | 43.93 | 16.81 | 76.14 |
| DFKI-MLT Contrastive (Transf.base) | 18.84 | 70.87 | 43.74 | 41.53 | 18.37 | 72.93 |
| DFKI-MLT Primary (Transf.large) | 18.86 | 70.98 | 43.79 | 41.53 | 18.41 | 72.95 |
| DFKI-MLT Contrastive (LSTM) | 19.88 | 69.35 | 44.28 | 40.91 | 19.43 | 71.36 |

Table 5: Results for the WMT18 APE **NMT subtask** – average TER (↓), BLEU score (↑). The symbol "†" indicates a difference from the MT baseline that is not statistically significant.

runs in terms of BLEU are different from those produced by the TER-based ranking). In both subtasks, however, the performance differences between the submitted runs are in general quite small: in a TER interval of less than one point we have the three top submissions to the PBSMT subtask and up to six submissions to the NMT subtask. In this situation, slightly different rankings produced by the two metrics are not surprising.

**PBSMT subtask.** This subtask has similar characteristics to the previous APE rounds. As shown by the results of the do-nothing baseline (24.24 TER, 62.99 BLEU), the original translations in the test set have a similar quality to those of the APE16 and APE17 En-De test sets (see Table 2). In spite of this, we observe further improvements compared to last year, in which the winning system was able to beat the baseline by -4.9 TER and +7.6 BLEU points. Also this year, all participants managed to beat the MT baseline at least with their primary submission but the top-ranked submission (MS_UEdin Primary) achieved larger improvements up to -6.24 TER and +9.53 BLEU points. Moreover, three submissions out of eleven outperformed the baseline by at least -5.0 TER and +8.0 BLEU points, which suggests a positive trend in terms of technology advancements. This can also be due to the availability of new additional training data (the eSCAPE corpus). However, verifying this hypothesis would require additional ablation tests since only one team (POSTECH) did not use all the available resources.

**NMT subtask.** In this subtask, the situation is rather different and the higher difficulty of correcting translations of better quality (16.84 TER, 74.73 BLEU) by learning from a smaller training set (less than half of the PBSMT subtask data) is confirmed. Results, even in the best case (FBK Primary), improve the baseline with a much smaller margin compared to the PBSMT subtask (-0.38 TER and +0.8 BLEU). Although they are obtained with the same neural technology successfully deployed for the PBSMT subtask, the majority of the scores fall in a range of less than one TER/BLEU point improvement over the baseline. Although not directly comparable, these results are in line with those of the APE17 evaluation, which was carried out on German-English phrase-based translations featuring a similar level of quality (15.55 TER, 79.54 BLEU, see Table 2). The fact that current neural APE technology performs similarly on phrase-based and neural outputs of comparable quality suggests that the quality of the machine-translated text to be corrected plays a more important role than the MT paradigm itself.

### 4.2 Automatic metrics computed using external references

By learning from (SRC, TGT, PE) triplets, APE systems' goal is to perform a "monolingual translation" from raw MT output into its correct version. In this translation process, the same sentence can be corrected in many possible ways that make the space of possible valid outputs potentially very large. Ideally, from this space, APE systems should select solutions that reflect as much as possible the post-editing style of the training data (in real-use settings, this can be the style/lexicon of specific users, companies, etc.). However, nothing prevents to end up with outputs that partially satisfy this constraint. In light of these considerations, TER and BLEU scores computed using human post-edits as reference represent a reliable measure of quality but:

1. They provide us with partial information on how systems' output reflects the post-editing style of the training data;

2. They are not informative at all about the amount of valid corrections that are not present in the human post-edits.

#### 4.2.1 Output style

To gain further insights on point 1., the "*TER (ref)*" and "*BLEU (ref)*" columns in Tables 4 and 5 show the TER and BLEU scores computed against independent reference translations. The rational behind their computation is that differences in "*TER/BLEU (pe)*" and "*TER/BLEU (ref)*" can be used as indicators of the "direction" taken by the trained models (i.e. either towards humans' post-editing style or towards a generic improvement of the MT output). Since independent references are usually very different from conservative human post-edits of the same TGT sentences, all the TER/BLEU scores measured using independent references are expected to be worse. However, if our hypothesis holds true, visible differences in the baseline improvements measured with "*TER/BLEU (pe)*" and "*TER/BLEU (ref)*" should indicate system's ability to model the post-editing style of the training data. In particular, larger gains measured with "*TER/BLEU (pe)*" will be associated to this desired ability.

**PBSMT subtask.** As can be seen in Table 4, the PBSMT subtask results show this tendency. Looking at the improvements over the baseline, those measured by computing TER and BLEU scores against human post-edits are often larger than those computed against independent references. In terms of TER, this holds true for the top six submitted runs, with the best system that shows a difference of 0.57 TER points in the gains over the baseline computed with "*TER (pe)*" (-6.24) and those computed with "*TER (ref)*" (-5.67). In terms of BLEU, the differences are more visible. For the top nine submissions, the baseline improvements are larger when computed with "*BLEU (pe)*". The best system improves over the baseline by 9.53 points with "*BLEU (pe)*" and 6.51 points with "*BLEU (ref)*", with a difference of 3.02 points that can be explained by its tendency to reflect the post-editing style of the training data.

**NMT subtask.** Similar considerations could be drawn for the NMT subtask but the small differences in the results reported in Table 5 (many of which are not statistically significant) do not allow to draw firm conclusions. For the top six submissions, TER and BLEU differences with respect to the baseline are larger when the two metrics are computed against post-edits. For the best submission, the improvements over the baseline are respectively 0.38 and 0.02 with "*TER (pe)*" and "*TER (ref)*". In terms of BLEU, they are 0.8 with "*BLEU (pe)*" and 0.08 with "*BLEU (ref)*".

### 4.2.2 Over-corrections

To shed light on point 2., the "*TER (pe+ref)*" and "*BLEU (pe+ref)*" columns in Tables 4 and 5 show the multi-reference TER and BLEU scores computed against post-edits and independent references. The rational behind their computation is that differences in "*TER/BLEU (pe)*" and "*TER/BLEU (pe+ref)*" can be used to analyze the quality of the unnecessary corrections performed by the systems (or, in other words, to study the impact of systems' tendency towards "over-correction"). APE corrections of a given MT output can indeed be of different types, namely: *i)* correct edits of a wrong passage, *ii)* wrong edits of a wrong passage, *iii)* correct edits of a correct passage and *iv)* wrong edits of a correct passage. TER/BLEU scores computed against human post-edits work reasonably well in capturing cases *i)-ii)* by matching APE systems' output with human post-edits: for wrong MT output passages (i.e. those changed by the post-editor), they inform us about the general quality of automatic corrections (i.e. how close they are to the post-editor's actions). Cases *iii)-iv)*, in contrast, are more problematic since any change performed by the system to a correct passage (i.e. those that were not changed by the post-editor) will always be penalized by automatic comparisons with human post-edits. Although discriminating between the two types of unnecessary corrections is hard, we hypothesize that a comparison between "*TER/BLEU (pe)*" and "*TER/BLEU (pe+ref)*" can be used as a proxy to quantify those belonging to type *iii)*. In general, due to the possibility to match more and longer n-grams in a multi-reference setting, "*TER/BLEU (pe+ref)*" scores are expected to be higher than "*TER/BLEU (pe)*" scores. However, if our hypothesis holds true, visible differences in the increase observed for the baseline and for the systems should indicate system's tendency to produce acceptable over-corrections (type *iii)*). In particular, larger gains observed for the APE systems will be associated to their over-correction tendency towards potentially acceptable edits that should not be penalized by automatic evaluation metrics.

**PBSMT subtask.** As can be seen in Table 4, the multi-reference results computed with "*TER/BLEU (pe+ref)*" are unsurprisingly better than those computed with "*TER/BLEU (pe)*". The variations of the do-nothing baseline are 0.48 TER points (from 24.24 with "*TER (pe)*" to 23.76 with "*TER (pe+ref)*") and 3.22 BLEU points (from 62.99 to 66.21). Interestingly, except for one system, all the results show larger variations when computed with "*BLEU (pe+ref)*", with a difference of 0.97 TER points (from 18.0 to 17.3) and 4.18 BLEU points (from 72.52 to 76.7) for the best system. Such variations are about 0.5 TER and 1.0 BLEU points larger than those measured for the baseline. This difference suggests that, though penalized by the comparison with human post-edits, a good amount of corrections made by the system still represent acceptable modifications of the original translations. Further analysis, which we leave for future work, should focus on understanding whether these corrections represent a problem (i.e. an unwanted deviation from the desired target style) or acceptable paraphrases of the input.

**NMT subtask.** Also in this case, as shown in Table 5, the multi-reference results computed with "*TER/BLEU (pe+ref)*" are better than those computed with "*TER/BLEU (pe)*". Apart from this, however, the performance variations for the baseline and the systems are not systematic nor particularly informative.

## 5 System/performance analysis

As a complement to global TER/BLEU scores, also this year we performed a more fine-grained analysis of the changes made by each system to the test instances.

### 5.1 Macro indicators: modified, improved and deteriorated sentences

Tables 6 and 7 show the number of modified, improved and deteriorated sentences, respectively for the PBSMT and the NMT subtasks. It's worth noting that, as in the previous rounds and in both the settings, the number of sentences modified by each system is higher than the sum of the improved and the deteriorated ones. This difference is represented by modified sentences for which the corrections do not yield TER variations. This grey area, for which quality improvement/degradation can not be automatically assessed, contributes to motivate the human evaluation discussed in Section 6.

**PBSMT subtask.** As can be seen in Table 6, the runs submitted to the PBSMT subtask reveal a quite homogeneous behaviour in terms of systems' aggressiveness. On average, the 11 submitted sys-

| Systems | Modified | Improved | Deteriorated |
|---|---|---|---|
| MS_UEdin Primary | 1,641 (82.05%) | 1,111 (67.70%) | 331 (20.17%) |
| FBK Contrastive (MRT+MLE) | 1,581 (79.05%) | 1,039 (65.72%) | 319 (20.18%) |
| FBK Primary (MRT) | 1,573 (78.65%) | 1,025 (65.16%) | 323 (20.53%) |
| POSTECH Contrastive (fix5) | 1,577 (78.85%) | 1,001 (63.47%) | 342 (21.69%) |
| POSTECH Primary | 1,566 (78.30%) | 992 (63.35%) | 338 (21.58%) |
| POSTECH Contrastive (var5) | 1,565 (78.25%) | 987 (63.07%) | 341 (21.79%) |
| USAAR_DFKI Primary | 1,435 (71.75%) | 751 (52.33%) | 469 (32.68%) |
| USAAR_DFKI* | 1,595 (79.75%) | 812 (50.91%) | 548 (34.36%) |
| DFKI-MLT Primary (Transf.large) | 1,221 (61.05%) | 469 (38.41%) | 457 (37.43%) |
| DFKI-MLT Contrastive (Transf.base) | 1,157 (57.85%) | 414 (35.78%) | 445 (38.46%) |
| DFKI-MLT Contrastive (LSTM) | 1,573 (78.65%) | 567 (36.05%) | 659 (41.89%) |

Table 6: Number of test sentences modified, improved and deteriorated by each run submitted to the **PBSMT subtask**.

| Systems | Modified | Improved | Deteriorated |
|---|---|---|---|
| FBK Primary (MRT) | 276 (26.98%) | 131 (47.46%) | 77 (27.90%) |
| MS_UEdin Primary | 316 (30.89%) | 150 (47.47%) | 107 (33.86%) |
| FBK Contrastive (MRT+MLE) | 298 (29.13%) | 134 (44.97%) | 88 (29.53%) |
| POSTECH Contrastive (top1) | 230 (22.48%) | 105 (45.65%) | 87 (37.83%) |
| POSTECH Primary (fix5) | 224 (21.90%) | 103 (45.98%) | 85 (37.95%) |
| POSTECH Contrastive (var5) | 220 (21.51%) | 101 (45.91%) | 85 (38.64%) |
| USAAR_DFKI Primary | 304 (29.72%) | 99 (32.57%) | 138 (45.39%) |
| DFKI-MLT Contrastive (Transf.base) | 468 (45.75%) | 60 (12.82%) | 351 (75.00%) |
| DFKI-MLT Primary (Transf.large) | 448 (43.79%) | 50 (11.16%) | 342 (76.34%) |
| DFKI-MLT Contrastive (LSTM) | 565 (55.23%) | 51 (9.03%) | 430 (76.11%) |

Table 7: Number of test sentences modified, improved and deteriorated by each run submitted to the **NMT subtask**.

tems modified about 75.0% of the sentences, with values ranging from 57.85% to 82.05%. In line with last year's round, the top-performing ones are more aggressive (the best systems peaks at 82.05% modified sentences) than those in lower-ranked positions. Since about 15.0% (i.e. 300) of the test instances are to be considered as "perfect" (see Figure1), the percentage of modifications is not too far to the expected value (85%). However, in terms of precision (i.e. the proportion of improved sentences out of the total amount of modified test items), the average is only 54.7%. While the three top submissions are able to improve more than 65.0% of the test items (with the best system peaking at 67.7%), the lower-ranked ones do not exceed 53.0%. The deteriorated sentences are on average 28.2%, with only three systems that are able to limit this proportion to about 20.0%. These results indicate that, although systems are able to change the expected number of sentences in the test set (with overall MT quality improvements, as shown in Table 4), their precision is still crucial. From this point of view, the room for improvement (more than 30 points in precision for the top submissions) remains large and advocates for solutions to drive APE technology towards the appropriate corrections (Chatterjee et al., 2018).

**NMT subtask.** In this subtask, the participating systems show a less aggressive behaviour and a tendency to preserve the higher quality of NMT translations. On average, the 10 submitted runs modified 32.7% of the sentences, with values ranging from 21.51% to 55.23%. However, though desirable, this behaviour is too conservative. Considering that about 25.2% (i.e. 257) of the test instances are to be considered as "perfect" (see Figure2), the reported numbers are far below the target percentage of modifications (74.8%). Also in terms of precision, the values are lower than in the PBSMT subtask. The average is 34.3% and even the top submissions have a percentage of improved sentences of less than 50.0%. The same holds for the percentage of deteriorated sentences (the average is 47.85%), for which all systems have larger values when dealing with neural outputs. Overall, the analysis confirms that correcting high-quality translations still remains a hard task, especially when dealing with NMT outputs. On

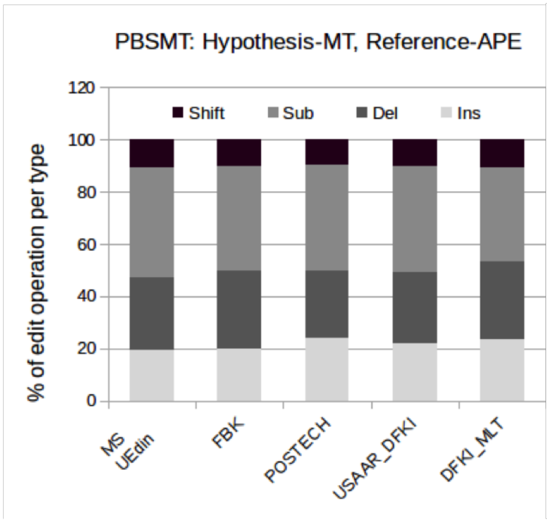


Figure 3: System behaviour (primary submissions) for the **PBSMT subtask** – TER(MT, APE)

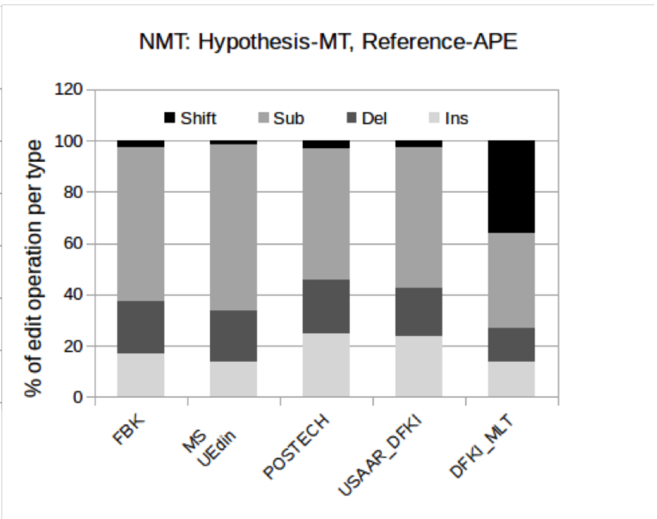


Figure 4: System behaviour (primary submissions) for the **NMT subtask** – TER(MT, APE)

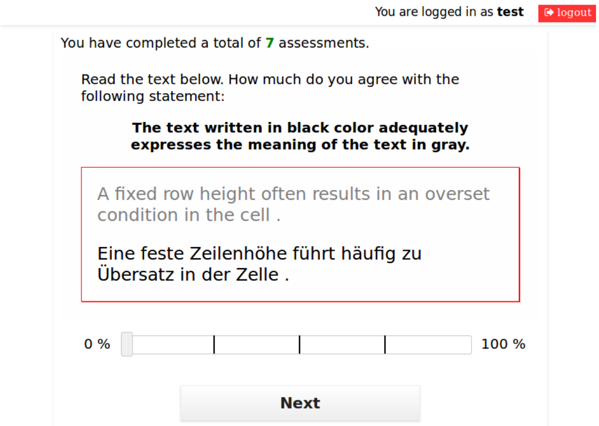
one side, as we observed in the PBSMT subtask, systems' low precision is an evident limitation. On the other side, and in addition to that, neural translations might be particularly difficult to improve, even for neural APE models. Since NMT is known to produce considerably less reordering errors than PBSMT (Bentivogli et al., 2016), one possible explanation is that the margins of improvement to the input sentences are reduced to types of errors (e.g. lexical) on which APE systems are less reliable. The analysis proposed in Section 5.2 aims to explore also this aspect.

### 5.2 Micro indicators: edit operations

We now turn to analyze the possible differences in the way systems corrected the test set instances. To this aim, we looked at the distribution of the edit operations done by each system (insertions, deletions, substitutions and shifts) by computing the TER between the original MT output and the output of each system taken as reference (only for the primary submissions). The outcomes of this analysis are shown in Figures 3 (PBSMT subtask) and 4 (NMT subtask).

**PBSMT subtask.** As it is evident from Figure 3, little can be said about the small differences in system's behaviour. Indeed, the plot does not show noticeable differences between neural-based submissions that, in most of the cases, implement similar solutions (multi-source, Transformer-based models trained with the same in-domain and artificial corpora). All of them are characterized by a rather homogeneous distribution of the types of correction patterns applied, with a large number of substitutions (average 39.8% of the total) and a slight dominance of deletions (average 28.2%) over the others (average insertions and shifts are respectively 21.6% and 10.4% of the total).

**NMT subtask.** Also in this case, most of the submissions are characterized by a similar behaviour, probably induced by the slightly different solutions adopted by participants. The distribution of edit operations, however, is less homogeneous than in the PBSMT subtask. Substitutions still represent the majority of the corrections but with a larger percentage (average 53.5%), which is followed by insertions (18.7%), deletions (18.5%) and shifts (9.2%). Average values, however, are influenced by one submission (DFKI-MLT), which shows a skewed distribution towards shift operations (36.15%) that are close in percentage to substitutions (36.88%). In terms of raw percentages, the role of shift operations can explain the lower performance of this outlier system, which was probably penalized by a large number of unnecessary reordering actions. As a more general observation, comparing Figures 3 and 4, we observe that reordering plays a quite different role in the two subtasks. Systems trained and evaluated on PBSMT data learn and apply more substitutions than those built for the NMT scenario. This can be explained by the higher fluency of neural translations which, among the four types of corrections, reduces the necessity of reordering operations. If this hypothesis holds true, the improvements of NMT outputs will mostly depend on other aspects like lexical choice, as suggested by the larger amount of substitutions compared to



You have completed a total of 7 assessments.

Read the text below. How much do you agree with the following statement:

**The text written in black color adequately expresses the meaning of the text in gray.**

A fixed row height often results in an overset condition in the cell .

Eine feste Zeilenhöhe führt häufig zu Übersatz in der Zelle .

Figure 5: Screenshot of the direct assessment user interface.

the PBSMT subtask.

# 6 Human evaluation

In order to complement the automatic evaluation of APE submissions, a manual evaluation of the primary systems submitted (five in total) was conducted. Similarly to the manual evaluation conducted for last year APE shared task, it was carried out following the direct assessment (DA) approach (Graham et al., 2013; Graham et al., 2016). In this Section, we present the evaluation procedure as well as the results obtained.

## 6.1 Evaluation procedure

The manual evaluation carried out this year involved 12 native German speakers with full professional proficiency in English in the IT domain, with a third of the evaluators being students in translation technologies from Saarland University and the remaining ones researchers and engineers from DFKI. Each evaluator was introduced to the evaluation task through a set of slides and a testing phase of the evaluation platform in order to be familiar with the user interface and its functionalities. A screenshot of the evaluation interface is presented in Figure 5.

A single assessment consists in assigning a score to a German sentence indicating how much of the meaning from a source sentence in English is expressed. In other words, the adequacy of a translation is directly evaluated on a scale from 0 to 100 given the source. The evaluators are free to conduct as many assessments as they want and free to schedule their own evaluation sessions. In addition, there was no requirement regarding a minimum amount of assessments to perform. The evaluation took place over a period of a month and was conducted in two sessions: a first one focusing on the PBSMT subtask and a second one on the NMT subtask.

| Subtask | PBSMT | NMT |
|---|---|---|
| # Systems | 7 | 7 |
| # Source segs | 2,000 | 1,023 |
| # Total Pairs | 14,000 | 7,161 |
| # Unique Pairs | 8,749 | 2,916 |
| Reduction | 37.5% | 59.3% |

Table 8: Data statistics per subtask with the total number of assessments prior to and after combination of identical target segments for each source.

For each subtask, the submitted post-edited test sets from the participants were presented to the evaluators one sentence at a time along with the corresponding source sentence. In order to define a baseline and an upperbound for this manual evaluation, the baseline (no post-edits) and the human post-edited MT output were added to the pool of submissions to evaluate, leading to a total of $14,000$ and $7,161$ pairs of segments to evaluate for the PBSMT and NMT subtasks respectively. However, it was possible to take advantage of the fact that multiple systems can produce identical outputs, allowing us to combine them and reduce the total number of source–target pairs to evaluate. Table 8 contains the statistics relative to the numbers of translations in total for all systems, as well as savings in terms of assessment effort that was gained by combining identical system outputs prior to running the evaluation.

Based on the direct assessment scores provided by the evaluators, two scores were computed for each system. A first score is the average of the segments direct assessment scores (noted ”Avg %”). For the second score (noted ”Avg $z$”), human assessments for translations were first standardized according to each individual human assessor's overall mean and standard deviation score. Average standardized scores for individual segments belonging to a given system are then computed, before the final overall DA score for that system is computed as the average of its segment scores.

## 6.2 Human Evaluation results

The twelve human evaluators spent a total of 64 hours on the DA task with an average of 17.2 and 17.5 seconds per assessment for the NMT and PBSMT subtasks respectively. More details about the

|  | # Assessments |  | Avg. Duration (sec.) |  |
|---|---|---|---|---|
| ID | PBSMT | NMT | PBSMT | NMT |
| 1 | 672 | 660 | 19.87 | 16.52 |
| 2 | 420 | 93 | 19.56 | 24.79 |
| 3 | 2,000 | 0 | 19.69 | - |
| 4 | 1,153 | 228 | 20.57 | 23.38 |
| 5 | 751 | 20 | 23.62 | 27.71 |
| 6 | 1,500 | 200 | 16.58 | 15.60 |
| 7 | 60 | 0 | 24.66 | - |
| 8 | 2,401 | 300 | 10.90 | 5.59 |
| 9 | 276 | 660 | 23.92 | 19.43 |
| 10 | 0 | 668 | - | 20.67 |
| 11 | 0 | 1,020 | - | 13.73 |
| 12 | 0 | 100 | - | 30.27 |

Table 9: Direct assessments statistics indicating the number of assessments carried out per subtask and the average duration in seconds per assessment for the twelve evaluators involved in the manual evaluation.

| # | Systems | Avg % | Ave $z$ |
|---|---|---|---|
|  | Human post-edit | 95.87 | 0.50 |
| 1 | MS_UEdin | 93.27 | 0.41 |
| 2 | FBK | 90.80 | 0.33 |
|  | POSTECH | 89.96 | 0.29 |
| 4 | USAAR_DFKI | 86.14 | 0.15 |
| 5 | DFKI-MLT | 77.78 | -0.15 |
|  | Baseline | 75.92 | -0.22 |

Table 10: DA Human evaluation results for the **PBSMT subtask** in terms of average raw DA (Ave %) and average standardized scores (Ave $z$). Dashed lines between systems indicate clusters according to Wilcoxon signed-rank test at p-level $p \leq 0.05$.

assessments done per evaluator, as well as the average duration per assessment, are presented in Table 9.

**PBSMT Subtask.** The results of DA for the PBSMT subtask are presented in Table 10. Six clusters are defined, grouping systems together according to which systems significantly outperform all others in lower ranking clusters based on the Wilcoxon signed-rank test. The human post-edited MT output reaches an averaged DA score of 95.87%, followed by the first system (MS_UEdin), single in a cluster and significantly better than the other systems, with an averaged DA score of 93.27%. A second cluster contains two systems which are non significantly different reaching 90.9% and 89.96% averaged DA scores.

All submitted systems are ranked significantly higher than the baseline (MT output without post-editing) but the top system remains below the human post-edits with a difference of 2.6%. The ranking of primary submissions for the PBSMT subtask is similar to the one obtained with the automatic metrics evaluation, where all primary systems were ranked above the baseline. For the DFKI-MLT system, TER indicates a non-significant difference with the baseline while DA scores leads to this system being significantly higher than the baseline.

| # | Systems | Ave % | Ave $z$ |
|---|---|---|---|
|  | Human post-edit | 96.13 | 0.43 |
| 1 | MS_UEdin | 91.11 | 0.24 |
|  | POSTECH | 90.41 | 0.22 |
|  | FBK | 90.41 | 0.20 |
|  | Baseline | 90.18 | 0.20 |
|  | USAAR_DFKI | 89.97 | 0.19 |
|  | DFKI-MLT | 89.53 | 0.18 |

Table 11: DA Human evaluation results for the **NMT subtask** in terms of average raw DA (Ave %) and average standardized scores (Ave $z$). Dashed lines between systems indicate clusters according to Wilcoxon signed-rank test at p-level $p \leq 0.05$.

**NMT Subtask.** The results of DA for the NMT subtask are presented in Table 11. Similarly to the results obtained with automatic metrics, the baseline is ranked above two and below three primary submissions. However, none of the submissions are ranked significantly higher or lower than the baseline according to DA scores and all five submissions are placed in the same cluster. The human post-edited MT output reaches an averaged DA score of 96.13%, ranked above the first system (MS_UEdin) with an averaged DA score of 91.11%.

The range of averaged DA scores for the NMT subtask is smaller ([89.53; 96.13]) compared to the PBSMT subtask ([75.92; 95.87]), which is observed in the results obtained with automatic metrics as well. This indicates a higher translation adequacy for the NMT subtask and is supported by the averaged DA score obtained by the baseline system (no post-edits). In addition, the human post-edited MT output reaches a higher averaged DA score for the NMT compared to the PBSMT subtask (similarly to automatic metrics results), which could indicate a higher overall translation quality of the final translation after manually post-editing the baseline NMT output compared to a baseline PBSMT output. However, more experiments involving larger test sets and a larger pool of evaluators are necessary to validate this observation.

## 7 Conclusion

We presented the results from the fourth shared task on Automatic Post-Editing. This year, we proposed two subtasks in which the MT output to be corrected was respectively generated by a phrase system (PBSMT subtask) and by a neural system (NMT subtask). Both the subtasks dealt with English-German data drawn from the information technology domain. This evaluation round attracted submissions from five groups, who submitted eleven runs for the PBSMT subtask and ten runs for the NMT one. Participants' systems have a lot in common: they are all neural models based on the Transformer architecture, some of them are based on multi-source methods and they all took advantage of the synthetic corpora released as additional training material. Evaluation results reflect such similarities and the effectiveness of the proposed solutions: top submissions have very close performance which, on both subtasks, shows significant improvements over the baseline.

In short, the main findings of this year's round are the following:

- Besides the amount of training data (the training corpora for the two subtasks have different size), the task difficulty is proportional to the quality of the initial translations. In line with previous years, learning from (and testing on) lower quality data leaves more room for improvement.
- The output of PBSMT systems is easier to improve (gains are up to -6.24 TER and +9.53 BLEU points). Such gains reflect a tendency to model the post-editors' style learned from training data.
- The output of NMT systems is harder to improve by current neural APE technology (gains are up to -0.38 TER and +0.8 BLEU points). A general explanations is that NMT translations are of higher quality. More specifically, looking the corrections done by the systems, the small number of reordering issues calls for effective methods to handle other types of errors (e.g. lexical choice) on which current APE technology can still be improved.
- Synthetic data help in improving performance. In the PBSMT subtask, similar to the APE17 English-German task from a task difficulty standpoint, the synthetic data provided as additional training material contributed to further improvements over the baseline.

## Acknowledgments

The manual evaluation conducted during the automatic post-editing task has been partially supported by the German Federal Ministry of Education and Research (BMBF) under the funding code 01IW17001 (Deeplee) and by the German Research Foundation (DFG) as part of SFB 1102 “Information Density and Linguistic Encoding”. We would like to thank the evaluators who participated in the manual evaluation and Christian Federmann for helping in data preparation and results extraction (Federmann, 2012).

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, Texas, November. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal, September. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark, September. Association for Computational Linguistics.

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the Planet of the APEs: a Comparative Study of State-of-the-art Methods for MT Automatic Post-Editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics)*, Beijing, China.

Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source Neural Automatic Post-Editing: FBK's Participation in the WMT 2017 APE Shared Task. In *Proceedings of the Second Conference on Machine Translation*, pages 630–638. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Frédéric Blain, and Lucia Specia. 2018. Combining Quality Estimation and Automatic Post-editing to Enhance Machine Translation Output. In *Conference of the American Association for Machine Translation (Research Track)*, New Orleans, LA.

Christian Federmann. 2012. Appraise: an open-source toolkit for manual evaluation of MT output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–36.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria, August. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, FirstView:1–28, 1.

Teresa Herrmann, Jan Niehues, and Alex Waibel. 2013. 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.

Chris Hokamp. 2017. Ensembling Factored Neural Machine Translation Models for Automatic Post-Editing and Quality Estimation. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September.

Matthias Huck, Fabienne Braune, and Alexander Fraser. 2017. Lmu munich's neural machine translation systems for news articles and health information texts. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 315–322, Copenhagen, Denmark, September. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation*, pages 751–758, Berlin, Germany, August.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. Microsoft and University of Edinburgh at WMT2018: Dual-Source Transformer for Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium, October.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. eSCAPE: a Large-scale Synthetic Corpus for Automatic Post-Editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018.

Santanu Pal, Nico Herbig, Antonio Krüger, and van Genabith Josef. 2018. A Transformer-Based Multi-Source Automatic Post-Editing System. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium, October.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318.

Daria Pylypenko and Raphael Rubino. 2018. DFKI-MLT System Description for the WMT18 Automatic Post-editing Task. In *Proceedings of the Third*

*Conference on Machine Translation*, Brussels, Belgium, October.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain, April.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692. Association for Computational Linguistics.

Jaehun Shin and Jong-Hyeok Lee. 2018. Multi-encoder Transformer Network for Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium, October.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical Phrase-Based Post-Editing. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT)*, pages 508–515, Rochester, New York.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

Lucia Specia, Kim Harris, Aljoscha Burchardt, Marco Turchi, Matteo Negri, and Inguna Skadina. 2017. Translation Quality and Productivity: A Study on Rich Morphology Languages. In *Proceedings of the 16th Machine Translation Summit*, Nagoya, Japan, September.

Amirhossein Tebbifakhr, Ruchit Agrawal, Matteo Negri, and Marco Turchi. 2018. Multi-source Transformer with Combined Losses for Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium, October.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

# Findings of the WMT 2018 Shared Task on Parallel Corpus Filtering

**Philipp Koehn**
Computer Science Department
Johns Hopkins University
Baltimore, Maryland, United States
phi@jhu.edu

**Huda Khayrallah**
Computer Science Department
Johns Hopkins University
Baltimore, Maryland, United States
huda@jhu.edu

**Kenneth Heafield**
School of Informatics
University of Edinburgh
Edinburgh, Scotland, European Union
kheafiel@inf.ed.ac.uk

**Mikel L. Forcada**
Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant
03690 St. Vicent del Raspeig, Spain
mlf@dlsi.ua.es

## Abstract

We posed the shared task of assigning sentence-level quality scores for a very noisy corpus of sentence pairs crawled from the web, with the goal of sub-selecting 1% and 10% of high-quality data to be used to train machine translation systems. Seventeen participants from companies, national research labs, and universities participated in this task.

## 1 Introduction

Training corpora for machine translation come in varying degrees of quality. On the one extreme end they are carefully professionally translated specifically for this purpose which may have done under the instruction to provide fairly literal translations and adherence to sentence-by-sentence correspondences. The other extreme are sentence pairs extracted with fully automatic processes from indiscriminate crawling of the World Wide Web.

The Shared Task on Parallel Corpus Filtering targets the second extreme, although the methods developed for this data condition should also carry over to less noisy parallel corpora. In setting this task, we were motivated by our ongoing efforts to create large publicly available parallel corpora from web sources and the recognition that noisy parallel data is especially a concern for neural machine translation (Khayrallah and Koehn, 2018).

This paper gives an overview of the task, presents its results and provides some analysis.

## 2 Related Work

Although the idea of crawling the web indiscriminately for parallel data goes back to the 20th century (Resnik, 1999), work in the academic community on extraction of parallel corpora from the web has so far mostly focused on large stashes of multilingual content in homogeneous form, such as the Canadian Hansards, Europarl (Koehn, 2005), the United Nations (Rafalovitch and Dale, 2009; Ziemski et al., 2015), or European Patents (Täger, 2011). A nice collection of the products of these efforts is the OPUS web site[1] (Skadiņš et al., 2014).

We are currently engaged in a large-scale effort to crawl text from the web. This work has been funded by Google Faculty Awards and is also currently funded by the European Union via the Connecting Europe Facility.[2] In 2016, we organized a shared task on document alignment as part of this effort (Buck and Koehn, 2016).

Acquiring parallel corpora from the web typically goes through the stages of identifying web sites with parallel text, downloading the pages of the web site, aligning document pairs, and aligning sentence pairs. A final stage of the processing pipeline filters out bad sentence pairs. These exist either because the original web site did not have any actual parallel data (garbage in, garbage out), or due to failures of earlier processing steps.

In 2016, a shared task on sentence pair filtering[3] was organized, albeit in the context of cleaning translation memories which tend to be cleaner that the data at the end of a pipeline that starts with web crawls.

There is a robust body of work on filtering out noise in parallel data. For example: Taghipour et al. (2011) use an outlier detection algorithm

[1] http://opus.lingfil.uu.se/

[2] http://www.paracrawl.eu/

[3] NLP4TM 2016: Shared task http://rgcl.wlv.ac.uk/nlp4tm2016/shared-task/

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 726–739
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64080

to filter a parallel corpus; Xu and Koehn (2017) generate synthetic noisy data (inadequate and nonfluent translations) and use this data to train a classifier to identify good sentence pairs from a noisy corpus; and Cui et al. (2013) use a graph-based random walk algorithm and extract phrase pair scores to weight the phrase translation probabilities to bias towards more trustworthy ones.

Most of this work was done in the context of statistical machine translation, but more recent work targets neural models. Carpuat et al. (2017) focus on identifying semantic differences in translation pairs using cross-lingual textual entailment and additional length-based features, and demonstrates that removing such sentences improves neural machine translation performance.

As Rarrick et al. (2011) point out, one type of noise in parallel corpora extracted from the web are translations that have been created by machine translation. Venugopal et al. (2011) propose a method to watermark the output of machine translation systems to aid this distinction. Antonova and Misyurev (2011) report that rule-based machine translation output can be detected due to certain word choices, and statistical machine translation output can be detected due to lack of reordering.

Belinkov and Bisk (2017) investigate the impact of noise on neural machine translation. They focus on creating systems that can *translate* the kinds of orthographic errors (typos, misspellings, etc.) that humans can comprehend. In contrast, Khayrallah and Koehn (2018) address noisy *training* data and focus on types of noise occurring in web-crawled corpora. They carried out a study how noise that occurs in crawled parallel text impacts statistical and neural machine translation.

There is a rich literature on data selection which aims at sub-sampling parallel data relevant for a task-specific machine translation system (Axelrod et al., 2011). van der Wees et al. (2017) find that the existing data selection methods developed for statistical machine translation are less effective for neural machine translation. This is different from our goals of handling noise since those methods tend to discard perfectly fine sentence pairs (say, about cooking recipes) that are just not relevant for the targeted domain (say, software manuals). Our task is focused on data quality that is relevant for all domains.

## 3 Task

The shared task tackled the problem of filtering parallel corpora. Given a noisy parallel corpus (crawled from the web), participants developed methods to filter it to a smaller size of high quality sentence pairs.

Specifically, we provided a very noisy 1 billion word (English token count) German–English corpus crawled from the web by the Paracrawl project. We asked participants to subselect sentence pairs that amount to (a) 10 million words, and (b) 100 million words, counted on the English side. The quality of the resulting subsets was determined by the quality of a statistical machine translation (Moses, phrase-based) and a neural machine translation system (Marian) trained on this data. The quality of the machine translation system was measured by BLEU score on the (a) official WMT 2018 news translation test set and (b) other undisclosed test sets.

Note that the task addressed the challenge of data quality and not domain-relatedness of the data for a particular use case. Hence, we discouraged participants from subsampling the corpus for relevance to the news domain. Thus, we place more emphasis on the undisclosed test sets, although we report both scores.

Participants in the shared task submitted a file with quality scores, one per line, corresponding to the sentence pairs. The scores do not have to be meaningful, except that higher scores indicate better quality. The scores were uploaded to a Google Drive folder which remains publicly accessible.[4]

Evaluation of the quality scores was done by subsampling 10 million and 100 million word corpora based on these scores, training statistical and neural machine translation systems with the subsampled corpora, and evaluation translation quality on blind test sets using the BLEU score.

For development purposes, we released configuration files and scripts that mirror the official testing procedure with a development test set. The development pack consists of

- a script to subsample corpora based on quality scores
- a Moses configuration file to train and test a statistical machine translation system
- Marian scripts to train and test a neural machine translation system

[4] https://drive.google.com/drive/folders/1zZNPlAThm-Rnvxsy8rXzChC49bc0_TGO

| Type of Noise | Count |
|---|---|
| Okay | 23% |
| Misaligned sentences | 41% |
| Third language | 3% |
| Both English | 10% |
| Both German | 10% |
| Untranslated sentences | 4% |
| Short segments (≤2 tokens) | 1% |
| Short segments (3–5 tokens) | 5% |
| Non-linguistic characters | 2% |

Table 1: Noise in the raw Paracrawl corpus.

- the test set from the WMT 2016 Shared Task on Machine Translation of News as development set
- the test set from the WMT 2017 Shared Task on Machine Translation of News as development test set

The web site for the shared task[5] provided detailed instructions on how to use these tools to replicate the official testing environment.

# 4 Data

## 4.1 Training Data

The provided raw parallel corpus is the outcome of a processing pipeline that aimed at high recall at the cost of precision, so it is very noisy. It exhibits noise of all kinds (wrong language in source and target, sentence pairs that are not translations of each other, bad language, incomplete of bad translations, etc.).

A cursory inspection of the corpus is given in Table 1. According to analysis by Khayrallah and Koehn (2018), only about 23% of the data is *okay*, but even that fraction may be flawed in some way. Consider the following sentence pairs that we did count as *okay* even though they contain mostly untranslated names and numbers.

*DE: Anonym 2 24.03.2010 um 20:55 314 Kommentare*
*EN: Anonymous 2 2010-03-24 at 20:55 314 Comments*

*DE: << erste < zurück Seite 3 mehr letzte >>*
*EN: << first < prev. page 3 next last >>*

It is an open question if such data is also harmful, merely irrelevant, or maybe even beneficial.

The raw corpus consists of a billion words of English, paired with German on the sentence level. It was deduplicated from a subset of the raw Paracrawl Release 1.

## 4.2 Provided Meta Information

The provided corpus file contains three items per line, separated by a TAB character:

- English sentence
- German sentence
- Hunalign score

The Hunalign scores were obtained from the sentence aligner (Varga et al., 2005). They may be a useful feature for sentence filtering, but they do not by themselves correlate strongly with sentence pair quality. None of the participants generally used this score.

Participant's systems may take the source of the data into account, e.g., by discounting sentence pairs that come from a web domain with generally low quality scores. To this end, we released the URL sources for each sentence pair as additional data set. Note that due to de-duplication a single sentence pair may have several URL pairs associated it, since it may appear on multiple web pages.

Participants were also allowed to use existing tools and external training data to build their filtering methods. Specifically, they were permitted to use the WMT 2018 news translation task data for German-English (without the Paracrawl parallel corpus) to train components of their method.

## 4.3 Test Sets

The goal of the task is to filter down to high-quality sentence pairs, but not to sentence pairs that are most fitting to a specific domain. During the submission period of the task, we only announced that we will use the official new translation test set from the WMT 2018 Shared Task of Machine Translation of News,[6] which was not released at that time yet.

In total, we used six test sets. For statistics see Table 2. Two of them were taken from existing evaluation campaigns, four were created for this shared task.

NEWSTEST2018 The test set from the WMT 2018 Shared Task of Machine Translation of

[5] http://www.statmt.org/wmt18/parallel-corpus-filtering.html

[6] http://www.statmt.org/wmt18/translation-task.html

News. It contains news stories that were either translated from German to English or from English to German.

IWSLT2017 The test set from the IWSLT 2017 evaluation campaign. It consists of transcripts of talks given at the TED conference. They cover generally accessible topics in the area of technology, entertainment, and design.

ACQUIS This test set was extracted from the Acquis Communtaire corpus, which is available on OPUS[7] (Tiedemann, 2012) (which was the source to create the subsequent 3 test sets). The test set consists of laws of the European Union that have to be incorporated into the national laws of the EU member countries. We only used sentences with 15 to 80 words, and removed any duplicate sentence pairs.

EMEA This test set was extracted from documents European Medicines Agency, which consist of public health announcements and descriptions of medications. We only used sentences with 20 to 80 words, and removed any duplicate sentence pairs.

GLOBALVOICES This test set was extracted from news stories posted and translated on Global Voices, an international and multilingual community of bloggers, journalists, translators, academics, and human rights activists. We selected several complete stories from this corpus.

KDE This test set was extracted from KDE4 localization files, which is open source software for Linux. We only used sentences with 15 to 80 words, and removed any duplicate sentence pairs.

For all the test sets, we checked for overlap with the training data, to prevent the possibility of having the test set being contained in the released noisy parallel data. We originally considered a test set based on the PHP documentation but removed it because that was contained in Paracrawl.

The official scoring of machine translation systems generated from the subsampled data sources is the average of the individual BLEU scores for each test set.

| **Test set** | Sentences | English Words |
|---|---:|---:|
| NEWSTEST2018 | 2998 | 58,628 |
| IWSLT2017 | 1138 | 18,162 |
| ACQUIS | 2862 | 98,624 |
| EMEA | 3000 | 93,071 |
| GLOBALVOICES | 3000 | 54,930 |
| KDE | 3000 | 109,716 |

Table 2: Statistics for the test sets used to evaluate the machine translation systems trained on the subsampled data sets. Word counts are obtained with `wc` on untokenized text.

## 5 Evaluation Protocol

The testing setup mirrors the development environment that we provided to the participants.

### 5.1 Particpants

We received submissions from 17 different organizations. See Table 3 for the complete list of participants. The participant's organizations are quite diverse, with 3 participants from Spain, 3 participants from the United States, 2 participants from Germany, 1 participant each from Canada, Greece, China, Japan, France, Latvia, Estonia, United Kingdom, and Brazil. 9 of the participants are companies, 3 are national research organizations, and 5 were universities.

Each participant submitted up to 5 different sets of scores, resulting in a total of 44 different submissions that we scored.

### 5.2 Subset Selection

We provided to the participants a file containing one sentence pair per line. A submission to the shared task consists of a file with the same number of lines, with one score per line corresponding to the quality of the corresponding sentence pair.

Using the score file, we selected subsets of a pre-defined size, defined by the number of English words. We chose the number of English words instead of German words, since the latter would allow selection of sentence pairs with very few German words and many English words which are beneficial for language model training but do not count much towards the German word total.

Subselecting sentence pairs is done by finding a threshold score, so that the sentence pairs that will be included in the subset have a quality score at and above this threshold. In some cases, a submission assigned this threshold score to a large num-

[7] http://opus.nlpl.eu/

| Acronym | Participant and System Description Citation |
|---|---|
| AFRL | Air Force Research Lab, USA (Erdmann and Gwinnup, 2018) |
| Alibaba | Machine Intelligence Technology Lab, Alibaba Group, China (Lu et al., 2018) |
| ARC | Inst. for Language and Speech Proc./Athena RC, Greece (Papavassiliou et al., 2018) |
| U Tartu | University of Tartu, Estonia (Barbu and Barbu Mititelu, 2018) |
| JHU | Johns Hopkins University, USA (Khayrallah et al., 2018) |
| LMU | Ludwig Maximilian University of Munich, Germany (Hangya and Fraser, 2018) |
| MAJE | WebInterpret, Spain (Fomicheva and González-Rubio, 2018) |
| Microsoft | Microsoft Corp., USA (Junczys-Dowmunt, 2018) |
| NICT | National Inst. of Information and Communications Tech., Japan (Wang et al., 2018) |
| NRC | National Research Council, Canada (Littell et al., 2018; Lo et al., 2018) |
| Prompsit | Prompsit, Spain (Sánchez-Cartagena et al., 2018) |
| RWTH | Rheinland-Westphälische Technical University, Germany (Rossenbach et al., 2018) |
| Speechmatics | Speechmatics, United Kingdom (Ash et al., 2018) |
| Systran | Systran, France (Pham et al., 2018) |
| Tilde | Tilde, Latvia (Pinnis, 2018) |
| UTFPR | Federal University of Technology, Paranà, Brazil (Paetzold, 2018) |
| Vicomtech | Vicomtech, Spain (Azpeitia et al., 2018) |

Table 3: Participants in the shared task.

ber of sentence pairs. Including all of them would yield a too large subset, excluding them yields a too small subset. Hence, we randomly included some of the sentence pairs to get the desired size in this case.

### 5.3 System Training

Given a selected subset of given size for a system submission, we built statistical (SMT) and neural machine translation (NMT) systems to evaluate the quality of the selected sentence pairs.

**SMT** For statistical machine translation, we used Moses (Koehn et al., 2007) with fairly basic settings, such as Good-Turing smoothing of phrase table probabilities, maximum phrase length of 5, maximum sentence length of 80, lexicalized reordering (*hier-mslr-bidirectional-fe*), fast-align for word alignment with *grow-diag-final-and* symmetrization, tuning with batch-MIRA, no operation sequence model, 5-gram language model trained on the English side of the subset with no additional data, and decoder beam size of 5,000.

**NMT** For neural machine translation, we used Marian (Junczys-Dowmunt et al., 2018). It uses the default settings of version 1.5, with 50,000 BPE operations, maximum sentence length of 100, layer normalization, dropout of 0.2 for RNN states, 0.1 for source embeddings and 0.1 for target embeddings, exponential smoothing, and decoding with beam size 12 and length normalization (1). Training a system for the 10 million word subset was limited to 20 epochs and took about 10 hours. Training a system for the 100 million word subset was limited to 10 epochs and took about 2 days.

Scores on the test sets were computed with `multi-bleu-detok.perl` included in Moses. We report case-insensitive scores.

## 6 Results

### 6.1 Core Results

The official results are reported in Table 4. The table contains the average BLEU score over all the 6 test sets for the 4 different setups

- statistical machine translation for 10 million word corpus
- statistical machine translation for 100 million word corpus
- neural machine translation for 10 million word corpus
- neural machine translation for 100 million word corpus

In the table, we highlight cells for the best scores for each of these settings, as well as scores that are close to it.

One striking observation is that the scores differ much more for the 10 million word subset than for

| **Participant** | **System** | **SMT 10M** | **SMT 100M** | **NMT 10M** | **NMT 100M** |
|---|---|---|---|---|---|
| AFRL | afrl-cvg-large | 21.9 | 25.2 | 13.8 | 30.2 |
| AFRL | afrl-cvg-mix-meteor | 23.4 | 25.3 | 27.1 | 30.3 |
| AFRL | afrl-cvg-mix | 22.5 | 25.2 | 19.8 | 30.1 |
| AFRL | afrl-cvg-small | 21.9 | 22.9 | 13.5 | 21.1 |
| AFRL | afrl-cyn-mix | 22.4 | 25.0 | 25.1 | 29.6 |
| Alibaba | alibaba-div | 24.1 | 26.4 | 27.6 | 31.9 |
| Alibaba | alibaba | 24.1 | 26.4 | 27.6 | 31.9 |
| ARC | arc-11 | 22.7 | 26.1 | 19.8 | 31.3 |
| ARC | arc-13 | 22.4 | 26.1 | 25.8 | 31.3 |
| ARC | arc-9 | 21.9 | 26.0 | 24.0 | 31.3 |
| U Tartu | tartu-hybrid-pipeline | 22.3 | 25.7 | 25.2 | 30.6 |
| JHU | zipporah-10000 | 22.6 | 25.8 | 25.3 | 30.2 |
| JHU | zipporah | 22.6 | 25.8 | 25.4 | 29.8 |
| LMU | lmu-ds-lm-si | 23.1 | 25.4 | 22.1 | 29.0 |
| LMU | lmu-ds-lm | 23.3 | 25.6 | 23.6 | 29.5 |
| LMU | lmu-ds | 23.3 | 25.5 | 23.6 | 29.5 |
| LMU | lmu | 21.5 | 25.6 | 23.0 | 30.5 |
| MAJE | webinterpet | 22.5 | 26.1 | 24.8 | 31.2 |
| Microsoft | microsoft | 24.4 | 26.5 | 28.6 | 32.1 |
| NICT | nict | 23.5 | 26.0 | 25.9 | 30.0 |
| NRC | nrc-mono-bicov | 21.0 | 26.2 | 23.1 | 31.6 |
| NRC | nrc-mono | 19.8 | 26.0 | 20.7 | 31.2 |
| NRC | nrc-seve-bicov | 22.1 | 26.2 | 25.3 | 31.7 |
| NRC | nrc-yisi-bicov | 23.9 | 26.4 | 27.4 | 31.9 |
| NRC | nrc-yisi | 23.5 | 26.4 | 26.5 | 31.8 |
| Prompsit | prompsit-al | 22.8 | 26.4 | 25.6 | 31.7 |
| Prompsit | prompsit-lm | 21.3 | 26.3 | 19.4 | 31.8 |
| Prompsit | prompsit-lm-nota | 20.1 | 26.2 | 19.3 | 31.7 |
| Prompsit | prompsit-sat | 22.9 | 26.3 | 26.1 | 31.7 |
| RWTH | rwth-count | 23.9 | 25.9 | 26.6 | 31.1 |
| RWTH | rwth-nn | 24.5 | 26.2 | 28.0 | 31.2 |
| RWTH | rwth-nn-redundant | 24.6 | 26.2 | 28.0 | 31.3 |
| Speechmatics | balanced-scoring | 23.8 | 25.8 | 27.9 | 31.0 |
| Speechmatics | prime-neural | 23.9 | 25.9 | 28.0 | 30.8 |
| Speechmatics | purely-neural | 18.1 | 25.8 | 18.0 | 30.0 |
| Systran | systran | 21.8 | 25.4 | 24.3 | 29.9 |
| Tilde | tilde-max-rescored | 23.0 | 26.0 | 26.6 | 31.2 |
| Tilde | tilde-max | 21.4 | 26.2 | 23.6 | 31.2 |
| Tilde | tilde-isolated | 21.0 | 25.9 | 22.6 | 30.8 |
| UTFPR | utfpr-tree | 17.6 | 20.7 | 11.4 | 11.9 |
| UTFPR | uftpr-regression | 20.8 | 22.4 | 21.8 | 22.2 |
| UTFPR | utfpr-forest | 13.2 | 17.0 | 6.6 | 6.2 |
| Vicomtech | vicomtech | 23.2 | 25.9 | 26.4 | 30.4 |
| Vicomtech | vicomtech-ngsat | 23.3 | 25.8 | 25.6 | 24.9 |

Table 4: Main results. BLEU scores (case-insensitive) are reported on the average of 6 test sets. Best performance on a test set is reported in bright green, scores within 0.5 BLEU points off the best in light green, and scores within 1 BLEU point off the best in light yellow.

the 100 million word subset. Scores also differ more for neural machine translation systems than for statistical machine translation systems.

For the 10 million word subset, there are only 2 submissions within 0.5 BLEU of the best system for statistical machine translation, and 0 for neural machine translation. For the 100 million word subset, there are 15 submissions within 0.5 BLEU of the best system for statistical machine translation, and 9 submissions within 0.5 for neural machine translation. Note that many of these submissions come from the same participants.

For both data sets, scores for neural machine translation are significantly higher. For the 10 million word subsets, the best NMT score is 28.6, while the best SMT score is 24.6. For the 100 million word subsets, the best NMT score is 32.1, while the best SMT score is 26.5. To be fair, statistical machine translation is typically trained with large monolingual corpora for language modelling that are essential for good performance.

### 6.2 Results by Test Set

Table 5 and 6 break out the results by each of the test sets, for statistical machine translation and neural machine translation, respectively.

The use of multiple test sets was motivated by the objective to discourage participants to filter sentence pairs for a specific domain, instead of filtering for general quality. Some participants used domain-specific data for training some elements of their filtering systems, such as monolingual news data sets to train language models but argued that these are broad domains that do not lead to domain over-fitting.

The results do not evoke the impression that some systems are doing better on some domains than others, at least not more than random variance would lead to expect. The closest test sets to the development sets are NEWSTEST2018, GLOBALVOICES, and maybe IWSLT2018. Only the 10 million word submissions *rwth-nn* and *rwth-nn-redundant* seem to do much better on these sets than others, relative to other submissions.

### 6.3 Additional Subset Sizes

Since we were interested in the shape of the curve of how different corpus sizes impact machine translation performance, we subselected additional subset size. Specifically, in addition to the 10 and 100 million word corpora, we also subselected 20, 30, 50, 80, 150, and 200 million words.

See Figure 1 for results for neural machine translation systems (also broken down by each individual test set) and Figure 2 for statistical machine translation systems. We only computed results for six systems due to the computational cost involved.

The scoring on additional subset sizes was not announced before the submission deadline for the shared task, so none of the participants optimized for these. In fact, some participants assigned the same low value for almost all sentence pairs that would be ignored when subselecting the 100 million word corpus. So, when subsampling larger corpora (150 and 200 million words, as we have done), the resulting system scores collapse.

The curves for neural machine translation system scores peak almost always at 100 million words, although also occasionally at 80 or 150 million words. Since we did not plot these curves when setting up the shared task, we cannot say if 100 million words is just a optimal value for this corpus or if participants overfitted their system to this value, although we would guess the first.

The performance between the submissions are quite similar on the different test sets. None of the submissions we show in the figures has overly optimized on the news test set.

## 7 Methods used by Participants

Not surprising due to the large number of submissions, many different approaches were explored for this task. However, most participants used a system using three components: (1) pre-filtering rules, (2) scoring functions for sentence pairs, and (3) a classifier that learned weights for feature functions.

**Pre-filtering rules.** Some of the training data can be discarded based on simple deterministic filtering rules. These may include rules to remove

- too short or too long sentences
- sentences that have too few words (tokens with letters instead of just special characters), either absolute or relative to the total number of tokens
- sentences whose average token length is too short or too long
- sentence pairs with mismatched lengths in terms of number of tokens

| Participant | System | 10M | | | | | | | 100M | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **AVERAGE** | NEWSTEST2018 | IWSLT2017 | ACQUIS | EMEA | GLOBALVOICES | KDE | **AVERAGE** | NEWSTEST2018 | IWSLT2017 | ACQUIS | EMEA | GLOBALVOICES | KDE |
| AFRL | afrl-cvg-large | 21.9 | 26.1 | 18.9 | 18.3 | 26.0 | 20.0 | 22.1 | 25.2 | 29.9 | 22.2 | 21.5 | 29.7 | 22.6 | 25.5 |
| AFRL | afrl-cvg-mix-meteor | 23.4 | 27.7 | 20.0 | 20.6 | 26.8 | 21.1 | 24.0 | 25.3 | 29.9 | 22.3 | 21.5 | 29.9 | 22.7 | 25.6 |
| AFRL | afrl-cvg-mix | 22.5 | 26.5 | 19.4 | 20.2 | 25.5 | 20.4 | 22.8 | 25.2 | 29.8 | 22.3 | 21.5 | 29.7 | 22.6 | 25.4 |
| AFRL | afrl-cvg-small | 21.9 | 26.2 | 18.9 | 18.3 | 26.0 | 20.1 | 22.1 | 22.9 | 27.1 | 20.0 | 20.9 | 25.8 | 21.2 | 22.4 |
| AFRL | afrl-cyn-mix | 22.4 | 26.6 | 19.5 | 19.7 | 25.7 | 20.3 | 22.8 | 25.0 | 29.4 | 22.2 | 21.3 | 29.5 | 22.4 | 25.3 |
| Alibaba | alibaba-div | 24.1 | 29.1 | 22.2 | 20.6 | 26.7 | 22.0 | 24.2 | 26.4 | 31.2 | 22.9 | 22.4 | 31.2 | 24.0 | 26.8 |
| Alibaba | alibaba | 24.1 | 28.9 | 22.1 | 20.5 | 26.8 | 22.0 | 24.2 | 26.4 | 31.1 | 23.0 | 22.5 | 31.2 | 24.0 | 26.8 |
| ARC | arc-11 | 22.7 | 26.9 | 18.9 | 19.3 | 27.2 | 20.4 | 23.3 | 26.1 | 30.8 | 22.7 | 22.4 | 30.9 | 23.5 | 26.6 |
| ARC | arc-13 | 22.4 | 26.3 | 18.8 | 18.7 | 26.5 | 20.2 | 23.8 | 26.1 | 30.6 | 22.8 | 22.3 | 30.9 | 23.4 | 26.7 |
| ARC | arc-9 | 21.9 | 26.0 | 18.1 | 18.5 | 25.8 | 20.0 | 23.2 | 26.0 | 30.7 | 22.7 | 22.1 | 30.9 | 23.4 | 26.3 |
| U Tartu | tartu-hybrid-pipeline | 22.3 | 26.8 | 19.5 | 18.7 | 24.8 | 20.6 | 23.5 | 25.7 | 30.4 | 22.3 | 21.9 | 30.5 | 23.1 | 26.1 |
| JHU | zipporah-10000 | 22.6 | 26.3 | 20.2 | 19.9 | 24.7 | 20.3 | 24.3 | 25.8 | 30.2 | 22.6 | 22.1 | 29.9 | 23.4 | 26.4 |
| JHU | zipporah | 22.6 | 26.3 | 20.4 | 19.3 | 24.8 | 20.4 | 24.3 | 25.8 | 30.4 | 22.6 | 22.1 | 30.1 | 23.3 | 26.5 |
| LMU | lmu-ds-lm-si | 23.1 | 27.6 | 20.8 | 17.7 | 26.6 | 21.5 | 24.4 | 25.4 | 30.0 | 22.3 | 21.4 | 29.9 | 23.1 | 26.0 |
| LMU | lmu-ds-lm | 23.3 | 28.0 | 20.6 | 18.0 | 26.9 | 21.4 | 24.7 | 25.6 | 30.1 | 22.4 | 21.5 | 30.1 | 23.1 | 26.2 |
| LMU | lmu-ds | 23.3 | 28.0 | 20.6 | 18.0 | 27.0 | 21.5 | 24.6 | 25.5 | 30.0 | 22.3 | 21.2 | 30.2 | 23.2 | 26.1 |
| LMU | lmu | 21.5 | 25.4 | 19.7 | 15.3 | 25.3 | 20.0 | 23.1 | 25.6 | 30.3 | 22.4 | 21.0 | 30.4 | 23.3 | 26.2 |
| MAJE | webinterpet | 22.5 | 27.2 | 21.3 | 19.1 | 24.5 | 21.2 | 22.0 | 26.1 | 30.7 | 22.9 | 22.4 | 30.6 | 23.7 | 26.2 |
| Microsoft | microsoft | 24.4 | 29.5 | 21.6 | 19.7 | 28.7 | 22.5 | 24.7 | 26.5 | 31.4 | 23.2 | 22.3 | 31.4 | 23.9 | 26.9 |
| NICT | nict | 23.5 | 27.8 | 20.9 | 19.3 | 25.9 | 21.4 | 25.5 | 26.0 | 30.8 | 22.8 | 22.0 | 30.4 | 23.4 | 26.6 |
| NRC | nrc-mono-bicov | 21.0 | 25.1 | 17.9 | 16.6 | 24.2 | 20.0 | 22.1 | 26.2 | 31.1 | 22.8 | 22.4 | 31.1 | 23.8 | 26.2 |
| NRC | nrc-mono | 19.8 | 23.5 | 16.6 | 15.5 | 23.1 | 18.6 | 21.4 | 26.0 | 30.6 | 22.7 | 22.1 | 30.7 | 23.7 | 26.2 |
| NRC | nrc-seve-bicov | 22.1 | 26.0 | 18.6 | 18.8 | 27.9 | 20.1 | 21.4 | 26.2 | 31.1 | 22.8 | 22.2 | 31.2 | 23.7 | 26.5 |
| NRC | nrc-yisi-bicov | 23.9 | 28.7 | 21.3 | 19.7 | 26.4 | 22.1 | 25.2 | 26.4 | 31.4 | 22.8 | 22.4 | 31.1 | 23.8 | 26.9 |
| NRC | nrc-yisi | 23.5 | 28.0 | 21.1 | 19.3 | 26.0 | 21.8 | 25.0 | 26.4 | 31.0 | 23.2 | 22.5 | 30.8 | 23.9 | 26.8 |
| Prompsit | prompsit-al | 22.8 | 26.0 | 19.9 | 19.1 | 27.0 | 20.1 | 24.3 | 26.4 | 31.2 | 22.8 | 22.5 | 31.3 | 23.8 | 26.9 |
| Prompsit | prompsit-lm | 21.3 | 25.4 | 19.5 | 16.9 | 23.2 | 19.3 | 23.3 | 26.3 | 31.1 | 22.8 | 22.5 | 31.0 | 23.6 | 26.6 |
| Prompsit | prompsit-lm-nota | 20.1 | 24.9 | 19.4 | 15.9 | 19.7 | 18.6 | 21.9 | 26.2 | 31.0 | 22.9 | 22.2 | 30.9 | 23.5 | 26.5 |
| Prompsit | prompsit-sat | 22.9 | 27.0 | 19.0 | 19.0 | 27.4 | 20.6 | 24.6 | 26.3 | 31.0 | 22.8 | 22.5 | 31.1 | 23.6 | 26.9 |
| RWTH | rwth-count | 23.9 | 28.6 | 21.8 | 21.0 | 26.8 | 22.0 | 22.8 | 25.9 | 30.7 | 22.9 | 22.0 | 30.2 | 23.5 | 26.3 |
| RWTH | rwth-nn | 24.5 | 29.6 | 21.8 | 21.4 | 28.0 | 22.7 | 23.8 | 26.2 | 30.8 | 23.2 | 22.2 | 30.9 | 23.4 | 26.6 |
| RWTH | rwth-nn-redundant | 24.6 | 29.6 | 21.8 | 21.4 | 28.1 | 22.6 | 23.9 | 26.2 | 30.8 | 23.1 | 22.1 | 30.9 | 23.6 | 26.8 |
| Speechmatics | balanced-scoring | 23.8 | 28.2 | 21.0 | 19.7 | 27.6 | 21.5 | 24.7 | 25.8 | 30.3 | 22.6 | 22.0 | 30.5 | 23.3 | 26.3 |
| Speechmatics | prime-neural | 23.9 | 28.2 | 20.5 | 19.6 | 28.3 | 21.4 | 25.3 | 25.9 | 30.4 | 22.5 | 21.9 | 30.7 | 23.3 | 26.4 |
| Speechmatics | purely-neural | 18.1 | 20.4 | 15.1 | 13.6 | 22.2 | 16.3 | 21.0 | 25.8 | 30.3 | 22.5 | 21.9 | 30.6 | 23.2 | 26.2 |
| Systran | systran | 21.8 | 25.4 | 19.4 | 16.7 | 25.7 | 19.9 | 23.9 | 25.4 | 30.0 | 22.3 | 21.5 | 30.1 | 22.7 | 26.1 |
| Tilde | tilde-max-rescored | 23.0 | 27.3 | 19.8 | 18.3 | 27.7 | 21.0 | 24.1 | 26.0 | 30.6 | 22.8 | 21.9 | 30.9 | 23.4 | 26.2 |
| Tilde | tilde-max | 21.4 | 25.0 | 18.2 | 16.6 | 25.6 | 19.7 | 23.6 | 26.2 | 30.8 | 22.8 | 22.1 | 31.1 | 23.6 | 26.6 |
| Tilde | tilde-isolated | 21.0 | 24.3 | 17.4 | 16.2 | 25.1 | 19.4 | 23.5 | 25.9 | 30.6 | 22.5 | 22.0 | 30.8 | 23.2 | 26.5 |
| UTFPR | utfpr-tree | 17.6 | 20.5 | 14.7 | 14.0 | 21.0 | 16.1 | 19.0 | 20.7 | 23.7 | 18.2 | 17.1 | 23.9 | 18.9 | 22.3 |
| UTFPR | uftpr-regression | 20.8 | 25.1 | 18.6 | 16.2 | 23.7 | 19.1 | 22.2 | 22.4 | 26.5 | 20.2 | 17.4 | 26.0 | 20.5 | 23.5 |
| UTFPR | utfpr-forest | 13.2 | 14.9 | 9.9 | 10.6 | 16.8 | 12.1 | 15.0 | 17.0 | 18.7 | 14.4 | 13.9 | 20.4 | 15.2 | 19.2 |
| Vicomtech | vicomtech | 23.2 | 27.5 | 20.4 | 19.3 | 26.5 | 21.2 | 24.6 | 25.9 | 30.5 | 22.5 | 22.2 | 30.3 | 23.4 | 26.6 |
| Vicomtech | vicomtech-ngsat | 23.3 | 27.5 | 19.8 | 19.3 | 26.8 | 21.1 | 25.1 | 25.8 | 30.2 | 22.4 | 22.1 | 30.0 | 23.4 | 26.7 |

Table 5: Detailed results for SMT performance. BLEU scores (case-insensitive) are reported on all the 6 test sets. The best performance on a test set is reported in bright green, scores within 0.5 BLEU points off the best in light green, and scores within 1 BLEU point off the best in light yellow.

| Participant | System | 10M | | | | | | | 100M | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **AVERAGE** | NEWSTEST2018 | IWSLT2017 | ACQUIS | EMEA | GLOBALVOICES | KDE | **AVERAGE** | NEWSTEST2018 | IWSLT2017 | ACQUIS | EMEA | GLOBALVOICES | KDE |
| AFRL | afrl-cvg-large | 13.8 | 11.2 | 6.1 | 15.5 | 23.8 | 8.9 | 17.4 | 30.2 | 37.0 | 26.3 | 26.5 | 35.1 | 28.0 | 28.2 |
| AFRL | afrl-cvg-mix-meteor | 27.1 | 33.4 | 23.3 | 25.6 | 29.9 | 25.4 | 25.0 | 30.3 | 37.4 | 26.0 | 26.6 | 35.2 | 28.1 | 28.4 |
| AFRL | afrl-cvg-mix | 19.8 | 19.7 | 10.9 | 23.9 | 26.9 | 14.8 | 22.7 | 30.1 | 37.4 | 26.1 | 26.4 | 34.8 | 28.1 | 28.1 |
| AFRL | afrl-cvg-small | 13.5 | 10.9 | 5.6 | 15.3 | 23.7 | 8.5 | 16.9 | 21.1 | 23.3 | 16.8 | 22.9 | 26.2 | 19.0 | 18.1 |
| AFRL | afrl-cyn-mix | 25.1 | 29.2 | 21.4 | 24.2 | 29.0 | 22.7 | 24.0 | 29.6 | 36.2 | 25.1 | 26.2 | 35.0 | 27.4 | 27.7 |
| Alibaba | alibaba-div | 27.6 | 35.0 | 25.2 | 24.1 | 29.8 | 25.8 | 25.7 | 31.9 | 39.5 | 27.1 | 28.4 | 36.7 | 29.1 | 30.7 |
| Alibaba | alibaba | 27.6 | 35.2 | 25.6 | 24.2 | 29.4 | 25.6 | 25.5 | 31.9 | 39.7 | 27.3 | 28.4 | 36.4 | 29.1 | 30.6 |
| ARC | arc-11 | 19.8 | 20.3 | 11.4 | 21.1 | 27.4 | 14.7 | 23.7 | 31.3 | 39.0 | 26.6 | 27.8 | 35.9 | 28.3 | 30.4 |
| ARC | arc-13 | 25.8 | 31.3 | 21.2 | 22.9 | 30.2 | 23.4 | 25.7 | 31.3 | 39.0 | 26.6 | 27.6 | 36.0 | 28.2 | 30.6 |
| ARC | arc-9 | 24.0 | 30.4 | 20.2 | 21.5 | 28.8 | 22.9 | 20.0 | 31.3 | 39.0 | 26.5 | 27.6 | 35.8 | 28.3 | 30.7 |
| U Tartu | tartu-hybrid-pipeline | 25.2 | 31.6 | 21.8 | 21.8 | 28.1 | 24.0 | 23.6 | 30.6 | 38.2 | 26.2 | 27.5 | 35.8 | 28.1 | 27.8 |
| JHU | zipporah-10000 | 25.3 | 31.4 | 23.1 | 22.8 | 26.3 | 24.0 | 24.3 | 30.2 | 36.8 | 24.2 | 27.6 | 35.4 | 27.7 | 29.3 |
| JHU | zipporah | 25.4 | 31.3 | 23.1 | 22.5 | 26.6 | 24.4 | 24.5 | 29.8 | 36.4 | 23.2 | 27.3 | 35.1 | 27.3 | 29.2 |
| LMU | lmu-ds-lm-si | 22.1 | 31.2 | 22.0 | 16.8 | 24.0 | 23.8 | 14.7 | 29.0 | 36.2 | 25.7 | 24.4 | 33.2 | 27.5 | 27.1 |
| LMU | lmu-ds-lm | 23.6 | 31.9 | 22.4 | 18.5 | 27.0 | 24.6 | 17.5 | 29.5 | 37.0 | 25.5 | 25.2 | 33.5 | 27.5 | 28.2 |
| LMU | lmu-ds | 23.6 | 31.8 | 22.1 | 18.4 | 27.1 | 24.5 | 17.9 | 29.5 | 36.7 | 25.5 | 25.2 | 34.1 | 27.7 | 27.9 |
| LMU | lmu | 23.0 | 28.8 | 21.1 | 16.0 | 27.0 | 23.3 | 21.6 | 30.5 | 37.8 | 25.9 | 25.8 | 35.6 | 28.5 | 29.6 |
| MAJE | webinterpet | 24.8 | 32.4 | 24.8 | 22.6 | 24.6 | 24.3 | 20.2 | 31.2 | 38.7 | 26.9 | 27.9 | 35.6 | 28.9 | 29.2 |
| Microsoft | microsoft | 28.6 | 35.7 | 25.1 | 23.7 | 32.7 | 26.7 | 27.8 | 32.1 | 39.9 | 27.4 | 28.3 | 36.7 | 29.3 | 30.8 |
| NICT | nict | 25.9 | 32.9 | 23.7 | 21.7 | 27.6 | 25.1 | 24.6 | 30.0 | 37.3 | 25.8 | 26.1 | 34.1 | 27.6 | 29.2 |
| NRC | nrc-mono-bicov | 23.1 | 27.9 | 19.3 | 19.0 | 26.4 | 22.0 | 23.7 | 31.6 | 38.9 | 27.1 | 28.1 | 36.0 | 28.9 | 30.4 |
| NRC | nrc-mono | 20.7 | 25.0 | 17.2 | 16.6 | 23.8 | 19.8 | 21.9 | 31.2 | 38.4 | 26.8 | 27.9 | 35.7 | 28.0 | 30.3 |
| NRC | nrc-seve-bicov | 25.3 | 30.3 | 21.5 | 22.6 | 31.7 | 23.1 | 22.9 | 31.7 | 39.4 | 27.1 | 28.3 | 36.3 | 28.9 | 30.1 |
| NRC | nrc-yisi-bicov | 27.4 | 33.9 | 24.4 | 23.2 | 29.8 | 25.4 | 27.8 | 31.9 | 39.6 | 26.9 | 28.4 | 36.6 | 29.1 | 30.7 |
| NRC | nrc-yisi | 26.5 | 32.7 | 23.9 | 22.2 | 28.6 | 24.8 | 26.8 | 31.8 | 39.3 | 27.1 | 27.9 | 36.3 | 29.0 | 30.9 |
| Prompsit | prompsit-al | 25.6 | 31.1 | 22.4 | 21.8 | 30.0 | 23.2 | 24.9 | 31.7 | 39.4 | 27.0 | 28.1 | 36.6 | 28.6 | 30.6 |
| Prompsit | prompsit-lm | 19.4 | 26.5 | 20.2 | 18.9 | 17.4 | 19.5 | 14.2 | 31.8 | 39.5 | 27.3 | 28.4 | 36.6 | 28.9 | 30.4 |
| Prompsit | prompsit-lm-nota | 19.3 | 26.1 | 20.0 | 18.8 | 17.3 | 19.8 | 14.0 | 31.7 | 39.8 | 26.7 | 28.3 | 36.4 | 29.1 | 30.0 |
| Prompsit | prompsit-sat | 26.1 | 31.6 | 20.8 | 22.1 | 31.2 | 23.7 | 26.8 | 31.7 | 39.2 | 26.7 | 28.2 | 36.4 | 28.7 | 30.8 |
| RWTH | rwth-count | 26.6 | 34.8 | 25.0 | 24.4 | 27.7 | 25.9 | 22.1 | 31.1 | 38.6 | 26.9 | 27.5 | 35.4 | 29.0 | 28.9 |
| RWTH | rwth-nn | 28.0 | 36.0 | 25.2 | 25.2 | 31.1 | 26.7 | 23.7 | 31.2 | 38.8 | 26.7 | 27.7 | 36.1 | 28.7 | 29.3 |
| RWTH | rwth-nn-redundant | 28.0 | 36.0 | 25.2 | 25.3 | 31.1 | 26.6 | 23.9 | 31.3 | 39.2 | 26.5 | 27.4 | 36.3 | 28.7 | 29.6 |
| Speechmatics | balanced-scoring | 27.9 | 34.0 | 24.6 | 24.7 | 30.9 | 25.0 | 28.0 | 31.0 | 37.8 | 26.5 | 27.9 | 35.4 | 28.2 | 30.1 |
| Speechmatics | prime-neural | 28.0 | 34.7 | 24.1 | 24.4 | 31.4 | 24.9 | 28.2 | 30.8 | 37.4 | 26.5 | 27.8 | 35.1 | 28.2 | 30.1 |
| Speechmatics | purely-neural | 18.0 | 21.8 | 15.6 | 13.1 | 21.3 | 17.6 | 18.4 | 30.0 | 35.2 | 25.8 | 26.9 | 35.1 | 27.4 | 29.8 |
| Systran | systran | 24.3 | 29.6 | 21.3 | 19.1 | 28.3 | 23.0 | 24.6 | 29.9 | 36.3 | 25.1 | 26.2 | 35.1 | 26.9 | 29.8 |
| Tilde | tilde-max-rescored | 26.6 | 32.4 | 22.1 | 22.1 | 31.3 | 24.4 | 27.1 | 31.2 | 38.6 | 26.8 | 27.5 | 36.6 | 28.2 | 29.6 |
| Tilde | tilde-max | 23.6 | 28.0 | 19.5 | 17.9 | 28.5 | 22.3 | 25.1 | 31.2 | 38.6 | 26.4 | 27.3 | 36.3 | 28.6 | 30.3 |
| Tilde | tilde-isolated | 22.6 | 26.6 | 18.9 | 16.8 | 27.4 | 21.8 | 24.2 | 30.8 | 38.0 | 25.8 | 26.7 | 35.7 | 27.9 | 30.4 |
| UTFPR | utfpr-tree | 11.4 | 13.2 | 7.8 | 10.4 | 17.5 | 9.9 | 9.8 | 11.9 | 10.5 | 6.8 | 11.7 | 18.2 | 10.1 | 13.9 |
| UTFPR | uftpr-regression | 21.8 | 27.2 | 18.5 | 18.6 | 24.9 | 19.2 | 22.1 | 22.2 | 25.0 | 16.7 | 19.1 | 28.8 | 19.7 | 24.1 |
| UTFPR | utfpr-forest | 6.6 | 6.5 | 2.9 | 4.2 | 11.5 | 5.9 | 8.3 | 6.2 | 4.7 | 2.1 | 3.5 | 12.3 | 5.0 | 9.3 |
| Vicomtech | vicomtech | 26.4 | 32.3 | 22.6 | 22.6 | 29.0 | 24.3 | 27.4 | 30.4 | 37.1 | 26.4 | 26.8 | 34.5 | 27.7 | 29.9 |
| Vicomtech | vicomtech-ngsat | 25.6 | 31.2 | 21.8 | 20.7 | 29.1 | 23.5 | 27.6 | 24.9 | 27.2 | 22.4 | 23.1 | 26.9 | 22.9 | 26.8 |

Table 6: Detailed results for NMT performance. BLEU scores (case-insensitive) are reported on all the 6 test sets. The best performance on a test set is reported in bright green, scores within 0.5 BLEU points off the best in light green, and scores within 1 BLEU point off the best in light yellow.

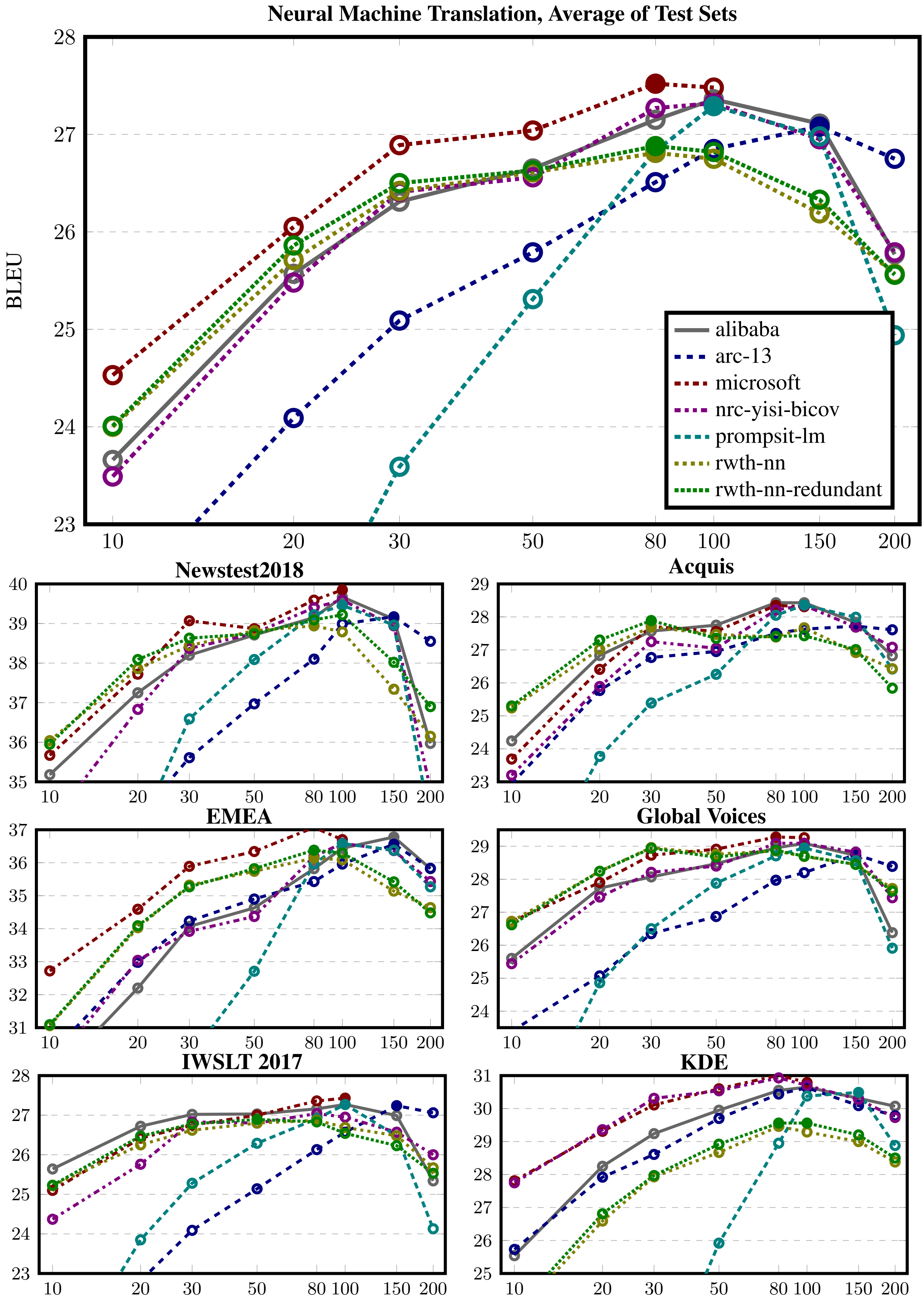


Figure 1: Additional corpus sizes, with breakdown by individual test set for some high-performing submissions. The charts plot BLEU scores against the size of the subselected corpus (in millions of words). The curves peak around 100 million words.

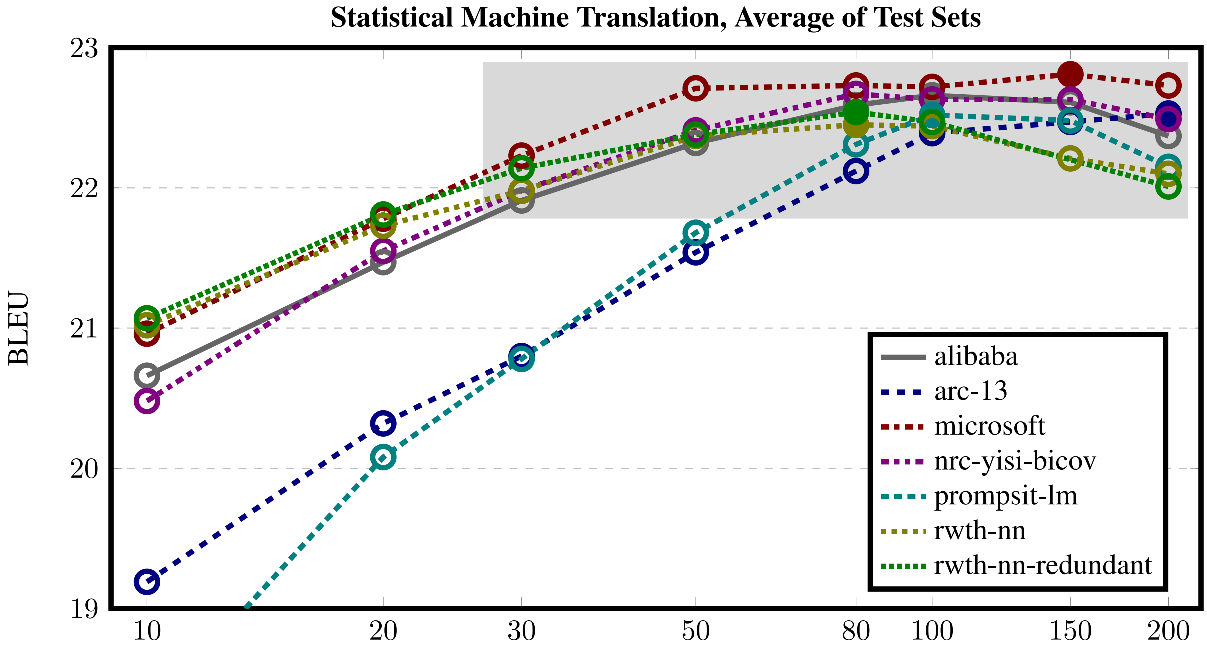


Figure 2: Version of Figure 1 for statistical machine translation systems built from the subselected data. Note that the curves are flatter, and the several systems score in a narrow band of 1 BLEU point across a wide range of corpus sizes (30-200 million words), indicated in grey.

- sentence pairs where names, numbers, email addresses, URLs do not match between both sides
- sentence pairs that are too similar, indicating simple copying instead of translating
- sentences where language identifier do not detect the required language

**Scoring functions.** Sentence pairs that pass the pre-filtering stage are assessed with scoring functions which provide scores that hopefully correlate with quality of sentence pairs. Participants used a variety of such scoring functions, including

- n-gram or neural language models on clean data
- language models trained on the provided raw data as contrast
- neural translation models
- bag-of-words lexical translation probabilities

Note that the raw scores provided by these models may be also refined in several ways. For instance, we may desire that the language model perplexities of a German sentence and its paired English sentence are similar. Or, we may contrast the translation model score for a sentence and its given paired sentence with the translation model score for the sentence and its best translation according to the model.

**Learning weights for scoring functions.** Given a large number of scoring functions, simply averaging their resulting scores may be inadequate. Learning weights to optimize machine translation system quality is computationally intractable due to the high cost of training these systems to evaluate different weight settings. A few participants used instead a classifier that learns how to distinguish between good and bad sentence pairs. Good sentence pairs are selected from existing high-quality parallel corpora, while bad sentence pairs are either synthesized by scrambling good sentence pairs or by using the raw crawled data.

Some participants made a distinction between unsupervised methods that did not use existing parallel corpora to train parts of the system, and supervise methods that did. Unsupervised methods have the advantage that they can be readily deployed for language pairs for which no seed parallel corpora exist.

## Acknowledgements

The shared task was supported by a Google Faculty Research Award to Johns Hopkins University and by the European Union through the Connected Europe Facility project *Provision of Web-*

*Scale Parallel Corpora for Official European Languages* (Paracrawl).

## References

Antonova, A. and Misyurev, A. (2011). Building a web-based parallel corpus and filtering out machine-translated text. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 136–144, Portland, Oregon. Association for Computational Linguistics.

Ash, T., Francis, R., and Williams, W. (2018). The speechmatics parallel corpus filtering system for wmt18. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Azpeitia, A., Etchegoyhen, T., and Martínez garcia, E. (2018). Stacc, oov density and n-gram saturation: Vicomtech's participation in the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Barbu, E. and Barbu Mititelu, V. (2018). A hybrid pipeline of rules and machine learning to filter web-crawled parallel corpora. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Belinkov, Y. and Bisk, Y. (2017). Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Buck, C. and Koehn, P. (2016). Findings of the wmt 2016 bilingual document alignment shared task. In *Proceedings of the First Conference on Machine Translation*, pages 554–563, Berlin, Germany. Association for Computational Linguistics.

Carpuat, M., Vyas, Y., and Niu, X. (2017). Detecting cross-lingual semantic divergence for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 69–79, Vancouver. Association for Computational Linguistics.

Cui, L., Zhang, D., Liu, S., Li, M., and Zhou, M. (2013). Bilingual data cleaning for SMT using graph-based random walk. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 340–345, Sofia, Bulgaria. Association for Computational Linguistics.

Erdmann, G. and Gwinnup, J. (2018). Coverage and cynicism: The afrl submission to the wmt 2018 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Fomicheva, M. and González-Rubio, J. (2018). Maje submission to the wmt2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Hangya, V. and Fraser, A. (2018). An unsupervised system for parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Junczys-Dowmunt, M. (2018). Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., and Birch, A. (2018). Marian: Fast neural machine translation in c++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Khayrallah, H. and Koehn, P. (2018). On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83. Association for Computational Linguistics.

Khayrallah, H., Xu, H., and Koehn, P. (2018). The jhu parallel corpus filtering systems for wmt 2018. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, Phuket, Thailand.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C. J., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Littell, P., Larkin, S., Stewart, D., Simard, M., Goutte, C., and Lo, C.-k. (2018). Measuring sentence parallelism using mahalanobis distances: The nrc unsupervised submissions to the wmt18 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Lo, C.-k., Simard, M., Stewart, D., Larkin, S., Goutte, C., and Littell, P. (2018). Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The nrc supervised submissions to the parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Lu, J., Lv, X., Shi, Y., and Chen, B. (2018). Alibaba submission to the wmt18 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Paetzold, G. (2018). Utfpr at wmt 2018: Minimalistic supervised corpora filtering for machine translation. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Papavassiliou, V., Sofianopoulos, S., Prokopidis, P., and Piperidis, S. (2018). The ilsp/arc submission to the wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Pham, M. Q., Crego, J., and Senellart, J. (2018). Systran participation to the wmt2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Pinnis, M. (2018). Tilde's parallel corpus filtering methods for wmt 2018. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Rafalovitch, A. and Dale, R. (2009). United Nations General Assembly resolutions: A six-language parallel corpus. In *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*. International Association for Machine Translation.

Rarrick, S., Quirk, C., and Lewis, W. (2011). MT detection in web-scraped parallel corpora. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 422–430. International Association for Machine Translation.

Resnik, P. (1999). Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Rossenbach, N., Rosendahl, J., Kim, Y., Graça, M., Gokrani, A., and Ney, H. (2018). The rwth aachen university filtering system for the wmt 2018 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Skadiņš, R., Tiedemann, J., Rozis, R., and Deksne, D. (2014). Billions of parallel words for free: Building and using the eu bookshop corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Sánchez-Cartagena, V. M., Bañón, M., Ortiz Rojas, S., and Ramírez, G. (2018). Prompsit's submission to wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Täger, W. (2011). The sentence-aligned european patent corpus. In Forcada, M. L., Depraetere, H., and Vandeghinste, V., editors, *Proceedings of the 15th International Conference of the European Association for Machine Translation (EAMT)*, pages 177–184.

Taghipour, K., Khadivi, S., and Xu, J. (2011). Parallel corpus refinement as an outlier detection algorithm. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 414–421. International Association for Machine Translation.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In Chair), N. C. C., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

van der Wees, M., Bisazza, A., and Monz, C. (2017). Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1411–1421. Association for Computational Linguistics.

Varga, D., Halaácsy, P., Kornai, A., Nagy, V., Németh, L., and Trón, V. (2005). Parallel corpora for medium density languages. In *Proceedings of the RANLP 2005 Conference*, pages 590–596.

Venugopal, A., Uszkoreit, J., Talbot, D., Och, F., and Ganitkevitch, J. (2011). Watermarking the outputs of structured prediction with an application in statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1363–1372, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Wang, R., Marie, B., Utiyama, M., and Sumita, E. (2018). Nict's corpus filtering systems for the wmt18 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Xu, H. and Koehn, P. (2017). Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2935–2940. Association for Computational Linguistics.

Ziemski, M., Junczys-Dowmunt, M., and Pouliquen, B. (2015). The united nations parallel corpus v1.0. In *International Conference on Language Resources and Evaluation (LREC)*.

# Meteor++: Incorporating Copy Knowledge into Machine Translation Evaluation

**Yinuo Guo, Chong Ruan, Junfeng Hu**[*]
Key Laboratory of Computational Linguistics, School of EECS, Peking University
{gyn0806, pkurc, hujf}@pku.edu.cn

## Abstract

In machine translation evaluation, a good candidate translation can be regarded as a paraphrase of the reference. We notice that some words are always copied during paraphrasing, which we call **copy knowledge**. Considering the stability of such knowledge, a good candidate translation should contain all these words appeared in the reference sentence. Therefore, in this participation of the WMT'2018 metrics shared task we introduce a simple statistical method for copy knowledge extraction, and incorporate it into Meteor metric, resulting in a new machine translation metric **Meteor++**. Our experiments show that Meteor++ can nicely integrate copy knowledge and improve the performance significantly on WMT17 and WMT15 evaluation sets.

## 1 Introduction

Automatic Metrics for machine translation (MT) evaluation have received significant attention in the past few years. MT evaluation measures how close machine-generated translations are to professional human translations, which can be treated as paraphrase evaluation except when the candidates are identical to references. The main difference is that MT evaluation only takes the correctness into consideration while paraphrase evaluation also focuses on diversity.

According to some previous studies on paraphrasing, we find that paraphrasing knowledge can be divided into two categories: copy knowledge and paraphrasable knowledge. The former reflects stable information which tends to keep intact during paraphrasing, while the latter can be paraphrased in various ways. There are some previous researches taking account of copy mechanism (Vinyals et al., 2015; Gu et al., 2016; See et al., 2017; Li et al., 2017) in text generation. And in this paper, we extend the idea of copy from generation to MT evaluation.

Firstly, we give an introduction to copy knowledge extraction on paraphrase corpus, and then propose Meteor++ incorporated with it based on Meteor. Our experiment results show that Meteor++ has higher Pearson Correlation with human score than Meteor on WMT evaluation sets and demonstrate the efficacy of copy knowledge.

## 2 Background

Various metrics for MT evaluation have been proposed and the widely used metrics are BLEU (Papineni et al., 2002) and Meteor (Banerjee and Lavie, 2005; Denkowski and Lavie, 2011, 2014). The main principle behind BLEU is the measurement of n-gram overlapping between the words produced by the machine and the human translation references at the corpus level. BLEU emphasizes precision and not take recall into account directly while Meteor not only combines the two but also gives a higher weight to recall in general. We choose Meteor in this paper because recall is extremely important for assessing the quality of MT output, as it reflects to what degree the translation covers the entire content of the source sentence.

The Meteor metric has been shown to have high correlation with human judgments in evaluation such as the 2010 ACL Workshop on Statistical Machine Translation and NIST Metrics MATR (Callison-Burch et al., 2010). It is based on general concept of flexible unigram matching, unigram precision and unigram recall, including the match of words that are simple morphological variants of each other by the identical stem and words that are synonyms of each other. Meteor firstly conduct an alignment include several stages (exact, stem, synonym and paraphrase) with different weight between two sentences. Then cal-

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 740–745
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64081

| word | c / p | word | c / p | word | c / p | word | c / p |
|---|---|---|---|---|---|---|---|
| instagram | 877/.950 | meth | 378/.923 | dandruff | 20/1.0 | communism | 21/1.0 |
| gmail | 725/.905 | python | 393/.908 | edmonton | 104/1.0 | algebra | 24/1.0 |
| traffic | 628/.936 | shotguns | 549/.961 | auckland | 104/1.0 | airprint | 97/1.0 |
| youtube | 621/.944 | linux | 173/.913 | vinegar | 31/1.0 | chess | 62/1.0 |
| java | 476/.901 | earthquake | 277/.981 | cellulite | 29/1.0 | officejet | 97/1.0 |
| kerala | 352/.989 | hacker | 267/.902 | hamsters | 75/1.0 | hamsters | 75/1.0 |
| macbook | 333/.931 | kvpy | 258/1.0 | bermuda | 63/1.0 | monday | 24/1.0 |
| sahara | 306/.935 | yahoo | 207/.913 | salman | 23/1.0 | forex | 36/1.0 |

Table 1: Quora “copy-words” examples, **c** means raw count and **p** means co-occurrence probability, totally we extract $427$ “ copy-words ” with $20$ as the **c** threshold and $0.85$ as the **p** threshold. Note that all the words are in their lower cases.

| word | c / p | word | c / p | word | c / p | word | c / p |
|---|---|---|---|---|---|---|---|
| president | 37/.833 | 10 | 27/.815 | 2016 | 13/1.0 | hamas | 4/1.0 |
| police | 36/.889 | women | 23/.913 | hepatitis | 3/1.0 | romania | 4/1.0 |
| world | 35/.886 | economy | 22/.867 | john | 3/1.0 | washington | 3/1.0 |
| russia | 34/.824 | government | 22/.818 | kingfisher | 5/1.0 | hundreds | 7/1.0 |
| million | 32/.813 | clinton | 22/.910 | garland | 9/1.0 | victim | 3/1.0 |
| trump | 31/.968 | thursday | 20/1.0 | local | 14/1.0 | facebook | 11/1.0 |
| putin | 18/1.0 | week | 17/.941 | ukraine | 9/1.0 | french | 7/1.0 |

Table 2: WMT “copy-words” examples, **c** means raw count and **p** means co-occurrence probability, we select the candidates with the human scores greater or equal to $0.7$ and combine them with their references as paraphrase pairs. Finally, we filter out $1088$ paraphrase pairs with a vocabulary of $4619$ words. Totally we extract $268$ “copy-words” with $2$ as **c** threshold and $0.8$ as the **p** threshold. Note that all the words are in their lower cases.

culate weighted precision $P$ and recall $R$. For each matcher ($m_i$), it counts the number of content and function words covered by matches of $ith$ type in the candidate ($m_i(h_c)$, $m_i(h_f)$) and reference ($m_i(r_c)$, $m_i(r_f)$), $|h_f|$ and $|r_f|$ mean the total number of function words in candidate and reference, $|h_c|$ and $|r_c|$ mean the total number of content words in candidate and reference.

$$P = \frac{\sum_i w_i \cdot (\delta \cdot m_i(h_c) + (1-\delta) \cdot m_i(h_f))}{\delta \cdot |h_c| + (1-\delta) \cdot |h_f|} \tag{1}$$

$$R = \frac{\sum_i w_i \cdot (\delta \cdot m_i(r_c) + (1-\delta) \cdot m_i(r_f))}{\delta \cdot |r_c| + (1-\delta) \cdot |r_f|} \tag{2}$$

The parameterized harmonic mean of precision $P$ and recall $R$ then calculated:

$$F_{mean} = \frac{P \cdot R}{\alpha \cdot P + (1-\alpha) \cdot R} \tag{3}$$

To account for gaps and differences in word order, a fragmentation penalty is calculated using the total number of matched words (m, averaged over hypothesis and reference) and number of chunks(ch):

$$Pen = \gamma \cdot (\frac{ch}{m})^{\beta} \tag{4}$$

The Meteor score is then calculated:

$$Score = (1 - Pen) \cdot F_{mean} \tag{5}$$

The parameters $\alpha$, $\beta$, $\gamma$, $\delta$ and $w_i...w_n$ are tuned to maximize correlation with human judgments.

## 3 Proposed Method

### 3.1 Copy Knowledge Extraction

According to our observation of paraphrasing corpus, we discover copy knowledge in which the words always have a high possibility of co-occurrence in paraphrase pairs. In this section, we will introduce a simple statistical method of copy knowledge extraction and present a word list denoted as **“copy-words”**. From this it can be concluded that if there is a missing “copy-word” in the candidate, it discards some important information; on the other hand, if the candidate contains any other extra “copy-words”, the two sentences

| categories | | examples | c / p |
|---|---|---|---|
| Named Entity | LOC | Sahara, Edmonton, Auckland, Russia, Romania, Washington | 62/8.9% |
| | ORG | WTO, OLA, PTE, MIT, HAI | 23/3.3% |
| | PER | Bob, Trump, Salman, Putin, John, Clinton | 123/17.7% |
| | MISC | Instagram, Gmail, communism, algebra, IQ, Monday, French, hundreds, million, 10, Linux, Python, Macbook, Yahoo, XBOX | 253/36.4% |
| OTHERS | | traffic, hacker, government, victim, economy | 234/33.7% |

Table 3: Copy knowledge classification, we combine the copy knowledge of Quora and WMT, and get 695 “copy-words” totally, **c** is the raw count and **p** is the proportion of each type.

are not semantically equivalent. Therefore the recall and precision of copy knowledge play a key role in the quality of translations.

In light of this, we propose a method for copy knowledge extraction in formula (6), $p_w$ means the co-occurrence probability, $C(w)$ means the raw appearance count of word and $C(co_w)$ means co-occurrence count. We select the words whose raw counts and co-occurrence probabilities in high-quality candidates and references exceed certain thresholds ($F$, $P$) as “copy-words”.

$$\text{“}copy\_words\text{”} = \{w\,|C(w) \geq F \;\wedge\; p_w \geq P\} \tag{6}$$

where

$$p_w = \frac{C(co_w)}{C(w)} \tag{7}$$

Here we test the method described above on the Quora[1] and the WMT datasets. The Quora dataset consists of over $400,000$ lines of potential question duplicate pairs. Each question pair has a binary value that indicates whether the line truly contains a duplicate pair. Here we only use the duplicate question pairs, including $142,963$ paraphrase pairs and a vocabulary of $32,582$ words. The WMT dataset consists of WMT15-17 (Bojar et al., 2017, 2016; Stanojević et al., 2015). We select the candidates with high human scores and combine them with their references as paraphrase pairs. There are 9287 pairs with human scores and only about one thousand pairs are useful. We regard the pairs which have human scores exceed the threshold as useful pairs (here we set the threshold as $0.8$). Since the amount of available texts with high human score is quite small, it is still not possible to conclude which words belong to copy knowledge.

Table 1 and Table 2 show part of the copy knowledge extraction results of the Quora and the WMT.

In Table 3, we divide the copy knowledge into several categories, and find that it is mainly composed of locations, persons, organizations, miscellaneousness and some others. We label these 695 $(427 + 268)$ “copy-words” manually and see that about $67\%$ of them are named entities. In general, named entity occupies a large proportion.

### 3.2 Model

Inspired by the observation of copy knowledge, we propose Meteor++ based on Meteor. In Meteor++, we incorporate copy knowledge into precision $P$ and recall $R$ indirectly. Specifically, we give penalties to the following two conditions from the perspective of recall and precision:

- **Recall :** there exist some “copy-words” only in references but not in candidates.
- **Precision :** there exist some “copy-words” only in candidates but not in references.

The candidates suffer the first condition may discard some important information, and the second may add some other extra information. We propose to correct the formulation of precision $P$ and recall $R$ in Meteor as following:

$$\tilde{P} = P \cdot \frac{X + \sum_i m_i(h_p)}{X + |h_p|} \tag{8}$$

$$\tilde{R} = R \cdot \frac{X + \sum_i m_i(r_p)}{X + |r_p|} \tag{9}$$

In formula (8), for each matcher $(m_i)$ , which counts the number of “copy-word” covered by matches of $i$-th type in the candidate $(m_i(h_p))$ and

[1] https://www.kaggle.com/quora/question-pairs-dataset

| lang-pair | de-en | fi-en | ru-en | ro-en | cs-en | tr-en | lv-en | zh-en |
|---|---|---|---|---|---|---|---|---|
| WMT17 | 2.102 | 1.776 | 2.251 | - | 1.892 | 2.201 | 2.232 | 2.772 |
| WMT16 | 1.833 | 1.988 | 2.065 | 2.148 | 1.499 | 2.357 | - | - |
| WMT15 | 1.621 | 1.816 | 1.876 | - | 1.492 | - | - | - |

Table 4: NE density of each language pair on WMT15-17, NE density means the average count of NE per sentence on each language pair.

|  | lang-pair | de-en | fi-en | ru-en | ro-en | cs-en | tr-en | lv-en | zh-en | avg |
|---|---|---|---|---|---|---|---|---|---|---|
| WMT2017 | Meteor | .535 | .719 | .618 | - | .550 | **.628** | .550 | .638 | .589 |
| ( X = 14 ) | Meteor++ | **.538** | **.720** | **.627** | - | **.552** | .626 | **.563** | **.646** | **.593** |
| WMT2015 | Meteor | .612 | .628 | **.622** | - | .582 | - | - | - | .600 |
| ( X = 6 ) | Meteor++ | **.626** | **.649** | **.622** | - | **.591** | - | - | - | **.609** |

Table 5: Segment-level Pearson correlation of Meteor and Meteor++ for to-English pairs on WMT15 and WMT17, where avg denotes the average Pearson correlation of all language pairs. The parameter $X$ in Meteor++ sets 14 on WMT17 and 8 on WMT15, other parameters are consist of the Meteor Universal.

the reference ($m_i(r_p)$), $|h_p|$ and $|r_p|$ respectively mean the total number of "copy-words" in the candidate and the reference. $X$ is a hyper-parameter used to smooth the results as following:

**For Smoothing :** In formula (1) and (2), we have already punished the unmatched words, here we only give an appropriate extra penalty to the "copy-words" missing.

**Compensation For The Gap :** In section 3.1, we only propose a simple statistical method to extract copy knowledge and it still has a long distance from the real copy knowledge.

Likewise, we have the modified recall formula as (9). After that correction, the $\tilde{P}$ and $\tilde{R}$ will substitute for the original $P$ and $R$ in the following calculation.

This two formulas can be regarded as using the precision and the recall of the "copy-words" to punish the entire sentence. If the "copy-words" are not identical in the candidate-reference pair, $P$ and $R$ will be discounted by the formula (8) and (9). We need to obtain a sufficiently high recall and precision of "copy-word" to guarantee the quality of the candidates since the copy knowledge is of greater importance.

## 4 Experiment Results

### 4.1 Settings

We evaluate our model on WMT15 and WMT17 metric task evaluation sets by calculating the correlation with the real human scores. The official human judgments of translation quality are collected using direct assessment(DA) (Graham et al., 2013). The direct assessment evaluation protocol give the annotators the reference and one MT output only and ask them to evaluate the translation adequacy of the MT output on an absolute scale.

The WMT datasets totally have $9287$ pairs with human scores and after filtering out the lower human score pairs, only about one thousand pairs can be regarded as the paraphrase pairs. As we described in section $3.1$, named entity is an important part of copy knowledge and accounts for $67\%$, here we take named entity as the copy knowledge because of the absence of reference-candidate pairs with high human scores on WMT datasets. And we use NLTK (Loper and Bird, 2002; Bird and Loper, 2004) toolkit to recognize named entities as our "copy-words" in experiments.

Table 4 shows the NE density of each language pair on WMT15-17 datasets and we select the WMT16 evaluation sets as our development sets. Our development experiments show that the parameter X has positive correlation with the NE density. We can see that WMT17 evaluation sets have higher NE density and WMT15 evaluation sets have lower NE density. In the experiments of Table 5, we set $X = 14$ on WMT17 and $X = 8$ on WMT15.

### 4.2 Results

Table 5 shows the Pearson correlation with the WMT15 and WMT17 direct assessment of translation adequacy at segment-level. We can see that Meteor++ has higher average segment-level Pearson correlation with DA human scores than Meteor on all WMT datasets.

## 5 Conclusion

In this paper, we describe the submissions of our metric Meteor++ for WMT18 Metrics task in detail. According to the observation of paraphrasing corpus, we discover copy knowledge in which the words keep intact after paraphrasing. We propose a simple statistical method to extract copy knowledge based on the given parallel monolingual paraphrases. Then, we present Meteor++ to examine the method of integrating copy knowledge into MT evaluation based on Meteor. Because words in copy knowledge always have a high possibility to be found in both candidates and references in machine translation, the Meteor++ could process better than Meteor. The experiment results on WMT datasets for each language pair show that Meteor++ has higher average segment-level Pearson correlation with DA human scores than Meteor and demonstrate the efficacy of copy knowledge.

## 6 Future Work

In this paper, we give a simple statistical method to extract copy knowledge, and propose the Meteor++ incorporate with it. Although it has already demonstrated great promise, we are still in the process of enhancing the metric in the following directions:

**Copy Knowledge Extraction:** We only propose a simple statistical method to extract copy knowledge which select the words with a high co-occurrence probability in paraphrase pairs. Here we just use bag-of-words to represent sentences and regard the intersection of them as co-occurrence. Therefore the copy knowledge we extract has a long way to go compared to the real copy knowledge. Furthermore, we are considering about constructing an alignment on the large-scale parallel monolingual corpus and then extracting universal copy knowledge based on it for broad use.

**Training the hyper-parameter $X$ on Data:** The hyper-parameter $X$ was designed to smooth the results and compensate for the gap between the copy knowledge we extract and the real copy knowledge. As our copy knowledge is getting more and more closer to the real copy knowledge, we plan to optimize the formulas by training on a separate data set, and choosing the $X$ formula with the best correlations with human assessment on the training data.

## References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Steven Bird and Edward Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the wmt16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 199–231.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar F Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the sixth workshop on statistical machine translation*, pages 85–91. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2017. Paraphrase generation with deep reinforcement learning. *arXiv preprint arXiv:1711.00279*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Miloš Stanojević, Amir Kamran, Philipp Koehn, and Ondřej Bojar. 2015. Results of the wmt15 metrics shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 256–273.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

# ITER: Improving Translation Edit Rate through Optimizable Edit Costs

**Joybrata Panja**[*]
Indian Institute of Technology (ISM)
Dhanbad, India
joybrata.15je001615@cse.ism.ac.in

**Sudip Kumar Naskar**
Jadavpur University
Kolkata, India
sudip.naskar@cse.jdvu.ac.in

## Abstract

The paper presents our participation in the WMT 2018 Metrics Shared Task. We propose an improved version of Translation Edit/Error Rate (TER). In addition to including the basic edit operations in TER, namely - insertion, deletion, substitution and shift, our metric also allows stem matching, optimizable edit costs and better normalization so as to correlate better with human judgement scores. The proposed metric shows much higher correlation with human judgments than TER.

## 1 Introduction

There has been several efforts to introduce better automatic evaluation metrics that can help towards the growth of machine translation (MT) systems. Human evaluation is slow and expensive and thereby efficient automatic MT evaluation metrics are required which are faster and correlate strongly with human judgements. Over the years a number of automatic MT evaluation metrics have been proposed like BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), Translation Edit Rate (Snover et al., 2002), NIST (Doddington, 2002), etc., which are widely used in the MT research and development community. However, due to its due to its simplicity and easier interpretability, Translation Edit Rate, or Translation Error Rate (TER), is one of the most commonly used MT evaluation metrics and often it is used as a baseline evaluation metric by MT researchers. In this work, we propose a new MT evaluation metric which provides improvements over TER and achieves better correlation with human judgement scores on the segment-level for various language pairs.

## 2 Related Work

The proposed metric is based on and an extension of TER (Snover et al., 2006), one of the most popular MT evaluation metrics. TER is an edit distance style error metric and it provides an edit ratio (often referred to as edit rate or error rate) in terms of how much editing is required to transform the MT output (also known as hypothesis) into a human translation (reference translation) with respect to the average length of the references. The term average is defined in case of multiple references, where normalization is done over the closest reference. The required editing is measured in terms of four edit operations - insertion, deletion, substitution and shifting.

Other related work relevant to our metric includes word error rate (WER) (Zechner and Waibel, 2000) and CharacTER (Wang et al., 2016). WER is the basis of TER and, unlike TER, it does not include the 'shift' operation. Both WER and TER consider word level edit operations. CharacTER is character level TER which calculates the edit distance at character level while performing the shift operations at word level.

Our work is different from CharacTER since we allow edit operations at character level only for those words in the hypothesis which find a stem match in the reference. Although TER outperforms WER, the normalization of the WER metric is the basis of our metric, i.e.,

[*] Work done while at Jadavpur University.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 746–750
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64082

normalization in our metric is a modified version of the normalization technique in WER.

## 3 Improvements over TER

Our metric includes all the edit operations carried out by TER, namely, insertion, deletion, substitution and shift. Apart from these operations, we improve over the TER metric by inclusion of stem matching, better normalization technique and optimal edit operation costs so as to improve the correlation of the new metric with human judgement score. We call our metric ITER, (*I*mproved *TER*).

### 3.1 Stemming cost

Stemming is a very standard technique widely used in many natural language processing tasks. Whenever a hypothesis word and a reference word are different while having the same stem, instead of substituting the entire word as in TER, we allow character level edit operations as follows:

**Hypothesis:** p l a y e d
**Reference:** p l a y i n g

**Edit operations:** nop(p), nop(l), nop(a), nop(y), sub(e, i), sub(d,n), ins(g)

Figure 1: Character level edit operations for tokens having the same stem. Here, nop refers to no-operation (i.e., character match), del(x) refers to deletion of character 'x', ins(x) refers to insertion of character 'x' and sub(x, y) refers to substitution of character 'x' by character 'y'.

In Figure 1, two substitutions and one insertion operation have to be made at character level in order to convert "played" into "playing". ITER uses Porter Stemmer available in the nltk package. Assuming that all edit operations have uniform cost of 1, we obtain the minimum edit cost of 3 for this string pair. The normalizing factor here is the number of the *'Edit operations:'* (cf. Figure 1) which includes the number of matched (corresponding to *nop*) characters plus the number of actual edit operations made. The motivation behind such normalization is to constrain the stemming cost to less than one. This is different from characTER and TER as their normalizing factors consider only characters and tokens of the reference respectively, and therefore exceeding their metric score over 1 (i.e., 100%) in case of number of insertions exceeding the hypothesis length.

$$\text{Stemming cost} = \frac{\min \mathit{edit\ cost}}{\#\mathit{characters\ matched} + \min \mathit{edit\ cost}}$$

### 3.2 Improved Normalization

Normalization at segment level is performed similar to the normalization for stem match. The minimum edit cost comprises of shifting cost, insertion cost, deletion cost, substitution cost and stemming cost (cf. Section 3.1). The normalization factor includes the total number of tokens (or words) in the hypothesis plus the number of tokens matched at the stem level and minimum edit cost.

$$\text{ITER} = \frac{\min \mathit{edit\ cost}}{\mathit{normalizing\ factor}}$$

where *normalizing factor* = #tokens of hypothesis + #tokens stemmed + min edit cost.

The first term (i.e. tokens of hypothesis) in the normalization factor represents that in the worst case, all tokens might need to be shifted. Considering the reference to be a different permutation (or alignment) of the tokens of the hypothesis with no added (or extra) tokens, we might go with shifting all the tokens of hypothesis. The shifted tokens could further be stemmed or edited thereby justifying the second and third component respectively in the normalization factor. In case stemming is not taken into consideration (as in the case of out-of-English translations, cf. Section 4.1), stemming cost is not be considered in min edit cost. Similarly, in the normalizing factor, there would not be any tokens stemmed, instead the concerned tokens will be substituted and will contribute to the min edit cost. Next we hypothesize that all edit costs lie between 0 and 1, therefore, in order to keep ITER in the [0, 1] range, we formulate our normalization in this way.

### 3.3 Optimal edit operation cost

TER considers equal cost for all the edit operations. The key motivation behind having optimal edit costs, or for that matter different edit costs for different edit operations, is that different edit operations take different time and effort during actual human post-editing. On the other hand, human judgement scores are direct

**Original Ref:** Hearts set for SFA battle over Neilson comments
**Original Hyp:** Hearts will fight SFA over comments against Neilson

**At cost 1:**2 shifts + 3 substitution
**Hyp After Shift:** Hearts will fight SFA against over Neilson comments → 2 shifts: [over, 1], [comments, 2]
**Edit operations after shift:** nop(Hearts), sub(will, set), sub(fight, for), nop(SFA), sub(against, battle), nop(over Nielson comments)

**At cost 2:**1 shift + 1 insertion + 1 deletion + 2 substitution
**Hyp After Shift:** Hearts will fight SFA over Neilson comments against → 1 shift: [Nielson, -2]
**Edit operations after shift:** nop(Hearts), sub(will, set), sub(fight, for), nop(SFA), ins(battle), nop(over Nielson comments), del(against)

Figure 2: Here, cost 1 and cost 2 represent two different cases of edit costs reflecting the change in edit operations. Although in both cases, there are 5 edit operations involved but the total edit cost will vary depending on the cost of each edit operation. The term [Nielson, -2] represents that 'Nielson' is shifted two places back. Similarly, [over, 1] represents that 'over' is shifted one place forward. For 'nop', 'ins', 'del' and 'sub', refer to section 3.1.

reflection of how much time and effort is required to correct the translations; they are inversely related. Prior to justifying the term 'optimal edit operation cost', Figure 2 explains the change in edit operations when the edit costs are changed. In order to find the optimal costs for the different edit operations, we trained our metric by varying each edit cost in the range [0.1, 1] with a difference of 0.1. Since we consider 4 different edit operations, this resulted in 10,000 (i.e., $10^4$) combinations for the edit costs. The set which gives the best correlation with human judgement scores is selected as the optimal set of edit costs.

## 4 Setup

ITER gives both segment level as well as system level score. Like TER, ITER is essentially a segment level metric and the system level scores are obtained by the weighted average of segment level scores. For optimizing the segment level scores, we trained our metric on the WMT15 datasets and tested on the WMT16 datasets for out-of-English and to-English translations.

### 4.1 Segment level score

Training data from WMT15 were used to tune our metric. All the edit operation costs were varied from 0 to 1 so as to find the optimal set of edit costs that results in highest correlation with human judgement scores.

For out-of-English translations, we skipped stemming since we could not avail reliable stemmers for the target languages and considered the basic operations at word level similar to TER. The normalizing factor of ITER have zero tokens to be stemmed in this case. Table 1 gives the optimal set of edit costs after training our metric on the WMT15 datasets.

### 4.2 System level score

The system level score is the weighted arithmetic average of segment level scores. Let us consider a test set having $m$ segments. We assume the ITERs to be $x_1, x_2, x_3, \ldots\ldots, x_m$for $m$ segments respectively where $x_i = e_i/n_i$. The term '$e_i$' represents minimum edit cost for the $i^{th}$ segment whereas '$n_i$' represents the normalizing factor for the $i^{th}$ segment. The system level score is defined as follows.

$$\mathrm{ITER}_{\mathrm{System}} = \frac{e_1 + e_2 + e_3 + \ldots\ldots\ldots + e_m}{n_1 + n_2 + n_3 + \ldots\ldots\ldots + n_m}$$

## 5 Experiments and Results

We tuned our metric on the training datasets of the WMT15 and obtained the following optimal sets of edit costs presented in Table 1.

| Lang_pair | D_cost | I_cost | Sh_cost | Sub_cost |
|---|---|---|---|---|
| cs-en | 0.5 | 0.7 | 0.3 | 0.9 |
| de-en | 0.7 | 0.4 | 0.5 | 1 |
| fi-en | 0.4 | 0.2 | 0.1 | 0.7 |
| ru-en | 0.5 | 0.3 | 0.1 | 0.6 |
| en-ru | 1 | 0.2 | 1 | 1 |

Table 1: Optimal sets of edit costs obtained after training ITER on WMT15 datasets (DAseg-wmt-newstest2015). Here, D_cost, I_cost, Sh_cost and Sub_cost refer to the cost of deletion, insertion, shifting and substitution, respectively.

We carried out the evaluation of our metric on the WMT16 (DAseg-wmt-newstest2016) dataset using the corresponding optimal sets of edit costs (cf. Table 1) tuned on the WMT15 datasets and computed the segment level correlation with human judgement scores in terms of Pearson correlation coefficient (Pearson, 1895). For a comparative evaluation, we compared our metric with TER on the same dataset and the results are shown in Table 2.

As can be seen from Table 2, the proposed metric provides much higher correlation (9.62% − 32.50%) for every language pair and target language than TER. The fact that even for the en-ru language direction, the metric shows significant improvement in correlation without the stem matching component, indicates that most of the improvements are due to the optimal edit costs. Apart from TER, we compared our results with the top performers of WMT16 segment level metrics (cf. Table 2) like sentBLEU (Bojar et al., 2016), MPEDA (Zhang et al., 2016) and METRICSF (Bojar et al., 2016). sentBLEU is the segment level version of BLEU, MPEDA was developed on the basis of METEOR and METRICSF is a combination of three metrics, namely, BLEU, METEOR and UPF-COBALT (Fomicheva et al., 2016). It can be inferred from Table 2 that ITER performs significantly better than TER and it is among the top few performers. Specifically for ru-en, ITER provides the best result and surpasses all other metrics.

We participated in the WMT 2018 Metrics Shared Task and submitted results for the “no hybrids” (newstest2018+testsuites) test set. Due to resource constraints, we could not evaluate the “hybrids” test set which contain artificially created 10K+ system outputs per language pair and test set. To establish better confidence intervals for system-level evaluation, the WMT18 metric task organizers computed system level scores for 10K hybrid super-sampled systems from our non-hybrid segment level scores using simple arithmetic average. The results of our participation in the WMT 2018 Metrics Shared Task are reported in (Ma et al., 2018).

## 6 Conclusions

This paper presents ITER, a TER style MT evaluation metric, which shows way better correlation than TER. The key idea behind ITER is optimizable edit costs. On the other hand, ITER gives the user the flexibility to choose their own set of edit operation costs and choose the one that suits the most. Since error rate higher than 100% does not make any sense, we improved the normalization in ITER. ITER also considers stem matching and character level edit operations.

## Acknowledgments

Sudip Kumar Naskar is supported by Digital India Corporation (formerly Media Lab Asia), MeitY, Government of India, under the Young Faculty Research Fellowship of the Visvesvaraya PhD Scheme for Electronics & IT.

## References

Satanjeev Banerjee, and Alon Lavie. 2005. An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *In Proceedings of the ACL-2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*. pages 65–72, Ann Arbor, Michigan.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the WMT16 Metrics Shared Task. *In Proceedings of the First Conference on Statistical Machine Translation*, pages 199–231, Berlin, Germany.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. *In Proceedings of the second international conference on Human Language Technology Research (HLT '02)*. Pages 138–145, San Diego, California.

Marina Fomicheva, Nuria Bel, Lucia Specia, Iria da Cunha, and Anton Malinovskiy. 2016. CobaltF: A Fluent Metric for MT Evaluation. *In Proceedings of the First Conference on*

| Lang_pair | ITER | TER | MPEDA | METRICSF | sentBLEU | DPMFCOMB | COBALTF | BEER |
|---|---|---|---|---|---|---|---|---|
| cs-en | 0.652 | 0.576 | 0.644 | 0.696 | 0.557 | 0.713 | 0.671 | 0.661 |
| de-en | 0.534 | 0.444 | 0.538 | 0.601 | 0.448 | 0.584 | 0.591 | 0.462 |
| fi-en | 0.524 | 0.478 | 0.513 | 0.557 | 0.484 | 0.598 | 0.554 | 0.471 |
| ru-en | 0.625 | 0.525 | 0.545 | 0.615 | 0.502 | 0.618 | 0.618 | 0.533 |
| en-ru | 0.591 | 0.446 | 0.645 | - | 0.550 | - | - | 0.666 |

Table 2: Segment-level correlations of automatic evaluation metrics on the WMT16 test set. Blank spaces indicate scores are not available. We calculated the ITER and TER scores and cited the other scores from the Bojar et al. (2016).

*Machine Translation,* Volume 2: Shared Task Papers, pages 483-490, Berlin, Germany.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 Metrics Shared Task. *In Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers,* Brussels, Belgium.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. 2002. BLEU: A method for automatic evaluation of machine translation. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (*ACL '02*), pages 311–318, Philadelphia, Philadelphia, PA.

Karl Pearson. 1895. Notes on Regression and Inheritance in the Case of Two Parents. *In Proceedings of the Royal Society of London*, volume 58, pages 240–242, London, UK.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. *In Proceedings of the Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA, USA.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, Hermann Ney. 2016. CharacTER: Translation Edit Rate on Character Level. *In Proceedings of the First Conference on Machine Translation,* Volume 2: Shared Task Papers, pages 505–510, Berlin, Germany.

Klaus Zechner and Alex Waibel. 2000. Minimizing Word Error Rate in Textual Summaries of Spoken Language. *In NAACL 2000 Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference,* Pages 186–193, Seattle, Washington, USA.

Lilin Zhang, Zhen Weng, Wenyan Xiao, Jianyi Wan, Zhiming Chen, Yiming Tan, Maoxi Li, and Mingwen Wang. 2016. Extract Domain-specific Paraphrase from Monolingual Corpus for Automatic Evaluation of Machine Translation. *In Proceedings of the First Conference on Machine Translation,* Volume 2: Shared Task Papers, pages 511-517, Berlin, Germany.

# RUSE: Regressor Using Sentence Embeddings for Automatic Machine Translation Evaluation

**Hiroki Shimanaka**[†] **Tomoyuki Kajiwara**[†‡] **Mamoru Komachi**[†]
[†]Graduate School of Systems Design, Tokyo Metropolitan University, Tokyo, Japan
shimanaka-hiroki@ed.tmu.ac.jp, komachi@tmu.ac.jp
[‡]Institute for Datability Science, Osaka University, Osaka, Japan
kajiwara@ids.osaka-u.ac.jp

## Abstract

We introduce the RUSE[1] metric for the WMT18 metrics shared task. Sentence embeddings can capture global information that cannot be captured by local features based on character or word N-grams. Although training sentence embeddings using small-scale translation datasets with manual evaluation is difficult, sentence embeddings trained from large-scale data in other tasks can improve the automatic evaluation of machine translation. We use a multi-layer perceptron regressor based on three types of sentence embeddings. The experimental results of the WMT16 and WMT17 datasets show that the RUSE metric achieves a state-of-the-art performance in both segment- and system-level metrics tasks with embedding features only.

## 1 Introduction

This study describes a segment-level metric for automatic machine translation evaluation (MTE). The MTE metrics with a high correlation with human evaluation enable the continuous integration and deployment of a machine translation (MT) system. Various MTE metrics have been proposed in the metrics task of the Workshops on Statistical Machine Translation (WMT) that was started in 2008. However, most MTE metrics are obtained by computing the similarity between an MT hypothesis and a reference based on the character or word N-grams, such as SentBLEU (Lin and Och, 2004), which is a smoothed version of BLEU (Papineni et al., 2002), Blend (Ma et al., 2017), MEANT 2.0 (Lo, 2017), and chrF++ (Popović, 2017). Therefore, they can exploit only limited information for the segment-level MTE. In other words, the MTE metrics based on character or word N-grams cannot make full use of sentence embeddings. They only check for word matches.

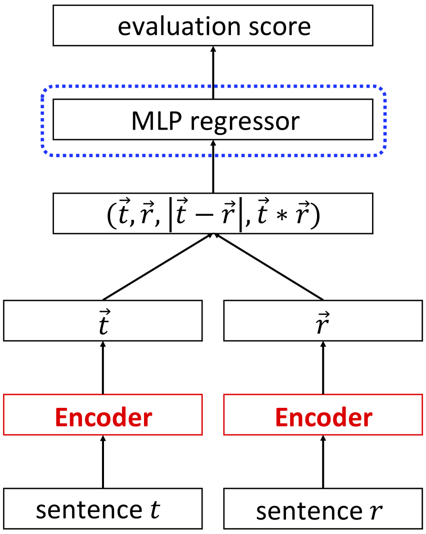


Figure 1: Outline of the RUSE metric.

We extend our previous work (Shimanaka et al., 2018) and propose a segment-level MTE metric using universal sentence embeddings capable of capturing global information that cannot be captured by local features based on character or word N-grams. The experimental results in both segment- and system-level metrics tasks conducted using the datasets for to-English language pairs on WMT16 and WMT17 indicated that the proposed regression model using sentence embeddings, RUSE, achieves the best performance.

The main contributions of the study are summarized below:

- We propose a novel supervised regression model for the segment-level MTE based on universal sentence embeddings.
- We achieved a state-of-the-art performance in segment- and system-level metrics tasks on the WNT16 and WMT17 datasets for to-English language pairs without using any complex features.

[1] https://github.com/Shi-ma/RUSE

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 751–758
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64083

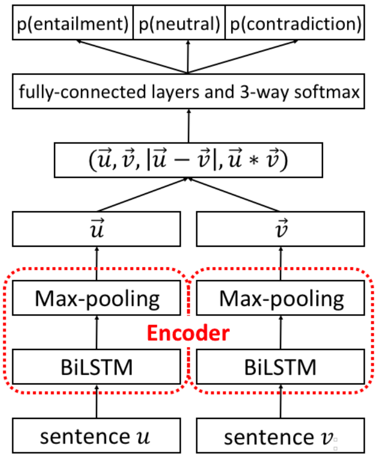


Figure 2: Outline of InferSent.

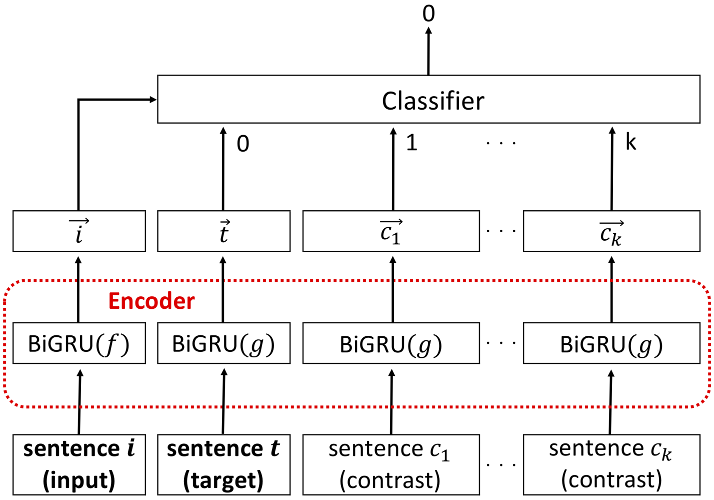

0
Classifier
0
1
· · ·
k
$\vec{i}$
$\vec{t}$
$\vec{c_1}$
$\vec{c_k}$
Encoder
BiGRU($f$)
BiGRU($g$)
BiGRU($g$)
BiGRU($g$)
sentence $\boldsymbol{i}$ (input)
sentence $\boldsymbol{t}$ (target)
sentence $c_1$ (contrast)
sentence $c_k$ (contrast)

Figure 3: Outline of Quick-Thought.

## 2 Related Work

DPMF$_{\text{comb}}$ (Yu et al., 2015a) achieved the best performance in the WMT16 metrics task (Bojar et al., 2016). It incorporates 55 default metrics provided by the Asiya MT evaluation toolkit[2] (Giménez and Màrquez, 2010), as well as three other metrics, namely DPMF (Yu et al., 2015b), REDp (Yu et al., 2015a), and ENTFp (Yu et al., 2015a), using ranking SVM to train parameters of each metric score. DPMF evaluates the syntactic similarity between an MT hypothesis and a reference translation. REDp evaluates an MT hypothesis based on the dependency tree of the reference translation that comprises both lexical and syntactic information. ENTFp (Yu et al., 2015a) evaluates the fluency of an MT hypothesis.

After the success of DPMF$_{\text{comb}}$, Blend[3] (Ma et al., 2017) achieved the best performance in the WMT17 metrics task (Bojar et al., 2017). Similar to DPMF$_{\text{comb}}$, Blend is essentially an SVR model with RBF kernel that uses the scores of various metrics as features. It incorporates 25 lexical metrics provided by the Asiya MT evaluation toolkit, as well as four other metrics, namely BEER (Stanojević and Sima'an, 2015), CharacTER (Wang et al., 2016), DPMF, and ENTFp. BEER is a linear model based on character N-grams and replacement trees. CharacTER evaluates an MT hypothesis based on character-level edit distance.

DPMF$_{\text{comb}}$ is trained through relative ranking (RR) of human evaluation data in terms of relative ranking (RR). The quality of five MT hypotheses of the same source segment is ranked from 1 to 5 via a comparison with the reference translation. In contrast, Blend is trained through direct assessment (DA) of human evaluation data. DA provides the absolute quality scores of hypotheses by measuring to what extent a hypothesis adequately expresses the meaning of the reference translation. The experiment results in the segment-level MTE conducted using the datasets for to-English language pairs on WMT16 showed that Blend achieved a performance better than DPMF$_{\text{comb}}$. In this study, as with Blend, we propose a regression model trained using DA human evaluation data.

Instead of using local and lexical features, ReVal[4] (Gupta et al., 2015a,b) proposes using sentence-level features. It is a metric using Tree-LSTM (Tai et al., 2015) for training and capturing the holistic information of sentences. It is trained using datasets of pseudo similarity scores generated by translating RR data and out-domain datasets of similarity scores of SICK[5]. However, the training dataset used in this metric consists of approximately 21,000 sentences; thus, the learning of Tree-LSTM is unstable, and accurate learning is difficult. We use sentence embeddings trained using various RNN and Transformer as sentence information. Furthermore, we apply universal sentence embeddings to this task. These embeddings were trained using large-scale data obtained in other tasks. Therefore, the proposed approach avoids the problem of using a small dataset for training sentence embeddings.

[2] http://asiya.lsi.upc.edu/

[3] http://github.com/qingsongma/blend

[4] https://github.com/rohitguptacs/ReVal

[5] http://clic.cimec.unitn.it/composes/sick.html

|  | cs-en | de-en | fi-en | lv-en | ro-en | ru-en | tr-en | zh-en |
|---|---|---|---|---|---|---|---|---|
| WMT15 | 500 | 500 | 500 | - | - | 500 | - | - |
| WMT16 | 560 | 560 | 560 | - | 560 | 560 | 560 | - |
| WMT17 | 560 | 560 | 560 | 560 | - | 560 | 560 | 560 |

Table 1: Number of segment-level DA human evaluation datasets for to-English language pairs[10] in WMT15 (Stanojević et al., 2015), WMT16 (Bojar et al., 2016), and WMT17 (Bojar et al., 2017).

|  |  | cs-en | de-en | fi-en | lv-en | ro-en | ru-en | tr-en | zh-en |
|---|---|---|---|---|---|---|---|---|---|
| WMT16 | systems | 6 | 10 | 9 | - | 7 | 10 | 8 | - |
|  | sentences | 2,999 | 2,999 | 3,000 | - | 2,998 | 1,999 | 3,000 | - |
| WMT17 | systems | 4 | 11 | 6 | 9 | - | 9 | 10 | 16 |
|  | sentences | 3,005 | 3,004 | 3,002 | 2,001 | - | 3,001 | 2,017 | 2,001 |

Table 2: Number of MT systems and system-level DA human evaluation datasets for to-English language pairs in WMT16 (Bojar et al., 2016) and WMT17 (Bojar et al., 2017).

## 3 RUSE: Regressor Using Sentence Enbeddings

The proposed metric evaluates the MT hypothesis with universal sentence embeddings trained using large-scale data obtained in other tasks. First, we describe three types of sentence embeddings used in the proposed metric in Section 3.1. We then explain the proposed regression model and feature extraction for MTE in Section 3.2.

### 3.1 Universal Sentence Embeddings

Several approaches have been proposed to learn sentence embeddings. These sentence embeddings are learned through large-scale data such that they constitute potentially useful features for MTE. These have been proven effective in various NLP tasks, such as document classification and measurement of semantic textual similarity, and we call them universal sentence embeddings.

First, InferSent[6] (Conneau et al., 2017) constructs a supervised model computing universal sentence embeddings trained using Stanford Natural Language Inference (SNLI) datasets[7] (Bowman et al., 2015). The Natural Language Inference task is a classification task of sentence pairs with three labels, namely *entailment*, *contradiction*, and *neutral*; thus, InferSent can train sentence embeddings that are sensitive to differences in meaning. This model encodes a sentence pair $u$ and $v$ and generates features by sentence embeddings $\vec{u}$ and $\vec{v}$ with a bi-directional LSTM architecture with max pooling (Figure 2). InferSent demonstrates high performance across various document classification and semantic textual similarity tasks.

Second, Quick-Thought[8] (Logeswaran and Lee, 2018) builds an unsupervised model of universal sentence embeddings trained using some consecutive sentences. Given an input sentence and its context, a classifier distinguishes context sentences from other contrastive sentences based on their embeddings (Figure 3). For a given sentence $s$, its embeddings are the concatenation of the outputs of the two encoders $[f(s); g(s)]$. As a result of the training, this encoder can produce sentence embedding. Quick-Thought demonstrates high performance, especially when applied to document classification tasks.

Finally, Universal Sentence Encoder[9] (Cer et al., 2018) is trained using multitask learning, whereby a single encoding model is used to feed multiple downstream tasks. Universal Sentence Encoder supports a task to estimate the neighboring sentences for unsupervised learning and tasks conversational input–response and natural language inference for supervised learning. The unsupervised learning model trained on data drawn from a variety of web sources, such as Wikipedia, web news, web question-answer pages and discussion forums, is augmented with training

[6] https://github.com/facebookresearch/InferSent

[7] https://nlp.stanford.edu/projects/snli/

[8] https://github.com/lajanugen/S2V

[9] https://www.tensorflow.org/hub/modules/google/universal-sentence-encoder-large/2

[10] en: English, cs: Czech, de: German, fi: Finnish, ro: Romanian, ru: Russian, tr: Turkish, lv: Latvian, zh: Chinese

| | cs-en | de-en | fi-en | ro-en | ru-en | tr-en | avg. |
|---|---|---|---|---|---|---|---|
| SentBLEU (Bojar et al., 2016) | 0.557 | 0.448 | 0.484 | 0.499 | 0.502 | 0.532 | 0.504 |
| COBALT-F (Bojar et al., 2016) | 0.671 | 0.591 | 0.554 | 0.639 | 0.618 | 0.627 | 0.617 |
| METRICS-F (Bojar et al., 2016) | 0.696 | 0.601 | 0.557 | 0.662 | 0.618 | 0.649 | 0.631 |
| DPMF$_{\text{comb}}$ (Bojar et al., 2016) | 0.713 | 0.584 | 0.598 | 0.627 | 0.615 | 0.663 | 0.633 |
| RUSE (MLP) with IS+QT+USE | 0.717 | **0.661** | **0.682** | **0.725** | 0.663 | 0.661 | **0.685** |
| RUSE (SVR) with IS+QT+USE | **0.720** | 0.632 | 0.678 | 0.708 | **0.670** | **0.675** | 0.681 |

Table 3: Segment-level Pearson correlation of metric scores and DA human evaluation scores for to-English language pairs in WMT16. IS: InferSent; QT: Quick-Thought; and USE: Universal Sentence Encoder.

| | cs-en | de-en | fi-en | lv-en | ru-en | tr-en | zh-en | avg. |
|---|---|---|---|---|---|---|---|---|
| SentBLEU (Bojar et al., 2017) | 0.435 | 0.432 | 0.571 | 0.393 | 0.484 | 0.538 | 0.512 | 0.481 |
| chrF++ (Bojar et al., 2017) | 0.523 | 0.534 | 0.678 | 0.520 | 0.588 | 0.614 | 0.593 | 0.579 |
| MEANT 2.0 (Bojar et al., 2017) | 0.578 | 0.565 | 0.687 | 0.586 | 0.607 | 0.596 | 0.639 | 0.608 |
| Blend (Bojar et al., 2017) | 0.594 | 0.571 | 0.733 | 0.577 | 0.622 | 0.671 | 0.661 | 0.633 |
| RUSE (MLP) with IS+QT+USE | 0.614 | 0.637 | **0.756** | **0.705** | **0.680** | 0.704 | 0.677 | 0.682 |
| RUSE (SVR) with IS+QT+USE | **0.624** | **0.644** | 0.750 | 0.697 | 0.673 | **0.716** | **0.691** | **0.685** |

Table 4: Segment-level Pearson correlation of metric scores and DA human evaluation scores for to-English language pairs in WMT17. IS: InferSent; QT: Quick-Thought; and USE: Universal Sentence Encoder.

on supervised data from the SNLI corpus. Universal Sentence Encoder demonstrates a higher performance across various document classification and semantic textual similarity tasks compared to InferSent.

### 3.2 Regression Model for MTE

This study proposes a segment-level MTE metric for to-English language pairs. This problem can be treated as a regression problem that estimates the translation quality as a real number from an MT hypothesis $t$ and a reference translation $r$. Once $d$-dimensional sentence vectors $\vec{t}$ and $\vec{r}$ are generated, the proposed model applies the following three matching methods to extract the relations between $t$ and $r$ (Figure 1).

- Concatenation: $(\vec{t}, \vec{r})$
- Element-wise product: $\vec{t} * \vec{r}$
- Absolute element-wise difference: $|\vec{t} - \vec{r}|$

Thus, we perform regression using $4d$-dimensional features of $\vec{t}$, $\vec{r}$, $\vec{t} * \vec{r}$ and $|\vec{t} - \vec{r}|$.

## 4 Experiments

We performed experiments using the evaluation datasets of the WMT metrics task to verify the performance of the proposed metric.

### 4.1 Setup

**Datasets.** We used segment-level datasets for to-English language pairs from the WMT15 (Stanojević et al., 2015), WMT16 (Bojar et al., 2016), and WMT17 (Bojar et al., 2017) metrics tasks as summarized in Table 1. For testing, we also used system-level datasets from the WMT16 and WMT17 metrics tasks as summarized in Table 2.

**Training.** We divided the dataset for training and development at a 9:1 ratio. First, for testing in WMT16, we divided the segment-level dataset of WMT15 into 1800 instances for training and 200 instances for development. Next, for testing in WMT17, we divided the segment-level datasets of WMT15 and WMT16 into 4824 instances for training and 536 instances for development. Finally, for submission to WMT18, we divided the segment-level dataset of WMT15, WMT16, and WMT17 into 8352 instances for training and 928 instances for development.

**Testing.** We scored each sentence using our metric for to-English language pairs in both segment and system levels. For testing on the system-level metrics task, we calculated the average score for each system as a system-level score. We evaluated our metric using the Pearson correlation coefficient between the metric scores and the DA hu-

|  | cs-en | de-en | fi-en | ro-en | ru-en | tr-en | avg. |
|---|---|---|---|---|---|---|---|
| BLEU (Bojar et al., 2016) | 0.989 | 0.808 | 0.864 | 0.840 | 0.837 | 0.895 | 0.872 |
| BEER (Bojar et al., 2016) | 0.990 | 0.879 | 0.972 | 0.852 | 0.901 | 0.982 | 0.929 |
| MPEDA (Bojar et al., 2016) | **0.993** | 0.937 | 0.976 | 0.932 | 0.929 | 0.982 | 0.958 |
| ReVal (Bojar et al., 2016) | 0.986 | **0.985** | 0.970 | 0.957 | **0.976** | 0.958 | 0.972 |
| RUSE (MLP) with IS+QT+USE | 0.990 | 0.968 | **0.977** | **0.962** | 0.953 | **0.991** | **0.974** |
| RUSE (SVR) with IS+QT+USE | 0.990 | 0.954 | 0.976 | 0.940 | 0.944 | 0.984 | 0.965 |

Table 5: System-level Pearson correlation of metric scores and DA human evaluation scores for to-English language pairs in WMT16. IS: InferSent; QT: Quick-Thought; and USE: Universal Sentence Encoder.

|  | cs-en | de-en | fi-en | lv-en | ru-en | tr-en | zh-en | avg. |
|---|---|---|---|---|---|---|---|---|
| BLEU (Bojar et al., 2017) | 0.971 | 0.923 | 0.903 | 0.979 | 0.912 | 0.976 | 0.864 | 0.933 |
| UHH_TSKM (Bojar et al., 2017) | **0.996** | 0.937 | 0.921 | 0.990 | 0.914 | 0.987 | 0.902 | 0.950 |
| BEER (Bojar et al., 2017) | 0.972 | 0.960 | 0.955 | 0.978 | 0.936 | 0.972 | 0.902 | 0.954 |
| Blend (Bojar et al., 2017) | 0.968 | **0.976** | 0.958 | 0.979 | **0.964** | 0.984 | 0.894 | 0.960 |
| RUSE (MLP) with IS+QT+USE | 0.995 | 0.964 | **0.985** | **0.996** | 0.956 | **0.993** | **0.937** | **0.975** |
| RUSE (SVR) with IS+QT+USE | **0.996** | 0.964 | 0.983 | 0.988 | 0.951 | **0.993** | 0.930 | 0.972 |

Table 6: System-level Pearson correlation of metric scores and DA human evaluation scores for to-English language pairs in WMT17. IS: InferSent; QT: Quick-Thought; and USE: Universal Sentence Encoder.

man scores.

**Features.** Publicly available pre-trained sentence embeddings, such as InferSent[6], Quick-Thought[8], and Universal Sentence Encoder[9], were used as the features mentioned in Section 3. InferSent is a collection of 4096-dimensional sentence embeddings trained on both 560,000 sentences of the SNLI dataset (Bowman et al., 2015) and 433,000 sentences of the MultiNLI dataset (Williams et al., 2018). Quick-Thought is a collection of 4800-dimensional sentence embeddings trained on both 45 million sentences of the BookCorpus dataset (Zhu et al., 2015) and 129 million sentences of the UMBC corpus (Han et al., 2013). Universal Sentence Encoder is a collection of 512-dimensional sentence embeddings trained on many sentences from a variety of web Sources, such as Wikipedia, web news, web question-answer pages, and discussion forums.

**Model.** Our regression model used a multi-layer perceptron (MLP) from Chainer[11] and Support Vector Regression (SVR) from sckit-learn[12] with the features mentioned in Section 3.2.

**MLP regressor.** Hyper-parameters were determined through grid search in the following parameters using the development data. We used ReLU as an activation function in all layers.

- Number of layers $\in \{1, 2, 3\}$
- Number of units $\in \{512, 1024, 2048, 4096\}$
- Batch size $\in \{64, 128, 256, 512, 1024\}$
- Dropout rate $\in \{0.1, 0.3, 0.5\}$
- Optimizer $\in \{\text{Adam}\}$

**SVR.** We used an SVR model with the RBF kernel. The hyper-parameters were determined through a 10-fold cross validation in the following parameters using the training and development data.

- $C \in \{0.1, 1.0, 10\}$
- $\epsilon \in \{0.01, 0.1, 1.0\}$
- $\gamma \in \{0.001, 0.01, 0.1\}$

**Baseline Metrics.** We compared the proposed metric with the four baseline metrics for each dataset. One is BLEU, which is the de facto standard metric for machine translation evaluation. The others are the top three metrics in each task.

[11] https://chainer.org/

[12] http://scikit-learn.org/

|  | cs-en | de-en | fi-en | lv-en | ru-en | tr-en | zh-en | avg. |
|---|---|---|---|---|---|---|---|---|
| Blend (Bojar et al., 2017) | 0.594 | 0.571 | 0.733 | 0.577 | 0.622 | 0.671 | 0.661 | 0.633 |
| RUSE (MLP) with IS+QT+USE | **0.614** | **0.637** | **0.756** | **0.705** | **0.680** | **0.704** | **0.677** | **0.682** |
| RUSE (MLP) with IS | 0.556 | 0.568 | 0.706 | 0.650 | 0.626 | 0.649 | 0.634 | 0.627 |
| RUSE (MLP) with QT | 0.601 | 0.587 | 0.737 | 0.685 | 0.661 | 0.692 | 0.647 | 0.658 |
| RUSE (MLP) with USE | 0.592 | 0.596 | 0.681 | 0.621 | 0.598 | 0.645 | 0.620 | 0.622 |

Table 7: Ablation analysis on the segment-level dataset in WMT17.

|  | cs-en | de-en | fi-en | lv-en | ru-en | tr-en | zh-en | avg. |
|---|---|---|---|---|---|---|---|---|
| Blend (Bojar et al., 2017) | 0.968 | **0.976** | 0.958 | 0.979 | **0.964** | 0.984 | 0.894 | 0.960 |
| RUSE (MLP) with IS+QT+USE | 0.995 | 0.964 | 0.985 | 0.996 | 0.956 | **0.993** | **0.937** | **0.975** |
| RUSE (MLP) with IS | 0.984 | 0.972 | 0.963 | 0.969 | 0.955 | 0.982 | 0.881 | 0.958 |
| RUSE (MLP) with QT | 0.997 | 0.952 | **0.997** | **0.998** | 0.945 | 0.992 | 0.936 | 0.974 |
| RUSE (MLP) with USE | **0.999** | 0.947 | 0.982 | 0.975 | 0.958 | 0.960 | 0.932 | 0.965 |

Table 8: Ablation analysis on the system-level dataset in WMT17.

### 4.2 Result

**Segment-level metrics task.** Tables 3 and 4 show the experimental results on the segment level. Our proposed metrics achieved the best performance in all to-English language pairs. For the segment-level tasks, both MLP and SVR regressors outperformed the state-of-the-art metrics.

**System-level metrics task.** Tables 5 and 6 present the experimental results on the system level. Our proposed metric based on the MLP regressor achieved the best performance in several to-English language pairs and outperformed the state-of-the-art metrics on average.

### 4.3 Discussion

These results indicated that adopting universal sentence embeddings in MTE is possible by training a regression model using DA human evaluation data. Blend is an ensemble method using combinations of various MTE metrics as features; hence, our results showed that universal sentence embeddings can more accurately consider the similarity between the MT hypothesis and the reference than a complex model.

**MLP vs. SVR in the RUSE metric.** These experimental results showed that in the RUSE metric, MLP performed better than SVR in many cases. In addition, MLP can be trained and inferred faster than SVR by making effective use of GPU. Therefore, we submitted a model of RUSE (MLP) with IS+QT+USE trained on the whole dataset to WMT18.

**Ablation analysis.** Tables 7 and 8 show that our metric with Quick-Thought feature only outperformed the state-of-the-art metrics in both segment- and system-level metrics tasks. Quick-Thought is an unsupervised model of universal sentence embeddings trained using some consecutive sentences. Therefore, Quick-Thought can be trained in corpora of languages other than English. Our method is effective if there are universal sentence embeddings and DA human evaluation data. Thus, our method with Quick-Thought may be effective in MTE for other than to-English language pairs.

## 5 Conclusions

In this study, we applied universal sentence embeddings to MTE based on the DA of human evaluation data. Our segment-level MTE metric RUSE achieved the best performance in both segment- and system-level metrics tasks on the WMT16 and WMT17 datasets. We conclude that:

- Universal sentence embeddings can more comprehensively consider information than an ensemble metric using combinations of various MTE metrics based on the features of character or word N-grams.
- Universal sentence embeddings trained on a large-scale dataset are more effective than sentence embeddings trained on a small or limited in-domain dataset.

## References

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 Metrics Shared Task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the WMT16 Metrics Shared Task. In *Proceedings of the First Conference on Machine Translation*, pages 199–231.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A Large Annotated Corpus for Learning Natural Language Inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *arXiv preprint arXiv:1803.11175v2*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Jesús Giménez and Lluís Màrquez. 2010. Asiya: An Open Toolkit for Automatic Machine Translation (Meta-) Evaluation. *The Prague Bulletin of Mathematical Linguistics*, (94):77–86.

Rohit Gupta, Constantin Orasan, and Josef van Genabith. 2015a. ReVal: A Simple and Effective Machine Translation Evaluation Metric Based on Recurrent Neural Networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1066–1072.

Rohit Gupta, Constantin Orasan, and Josef van Genabith. 2015b. Machine Translation Evaluation using Recurrent Neural Networks. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 380–384.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 44–52. Association for Computational Linguistics.

Chin-Yew Lin and Franz Josef Och. 2004. ORANGE: a Method for Evaluating Automatic Evaluation Metrics for Machine Translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 501–507.

Chi-Kiu Lo. 2017. MEANT 2.0: Accurate Semantic MT Evaluation for Any Output Language. In *Proceedings of the Second Conference on Machine Translation*, pages 589–597.

Lajanugen Logeswaran and Honglak Lee. 2018. An Efficient Framework for Learning Sentence Representations. In *International Conference on Learning Representations*.

Qingsong Ma, Yvette Graham, Shugen Wang, and Qun Liu. 2017. Blend: a Novel Combined MT Metric Based on Direct Assessment —CASICT-DCU submission to WMT17 Metrics Task. In *Proceedings of the Second Conference on Machine Translation*, pages 598–603.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Maja Popović. 2017. chrF++: Words Helping Character N-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. Metric for Automatic Machine Translation Evaluation based on Universal Sentence Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 106–111.

Miloš Stanojević, Philipp Koehn, and Ondřej Bojar. 2015. Results of the WMT15 Metrics Shared Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 256–273.

Miloš Stanojević and Khalil Sima'an. 2015. BEER 1.1: ILLC UvA Submission to Metrics and Tuning Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 396–401.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1556–1566.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTER: Translation Edit Rate on Character Level. In *Proceedings of the First Conference on Machine Translation*, pages 505–510.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational*

*Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Hui Yu, Qingsong Ma, Xiaofeng Wu, and Qun Liu. 2015a. CASICT-DCU Participation in WMT2015 Metrics Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 417–421.

Hui Yu, Xiaofeng Wu, Wenbin Jiang, Qun Liu, and Shouxun Lin. 2015b. An Automatic Machine Translation Evaluation Metric Based on Dependency Parsing Model. *arXiv preprint arXiv:1508.01996*.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

# Keep It or Not: Word Level Quality Estimation for Post-Editing

**Prasenjit Basu[1], Santanu Pal[2,3], Sudip Kumar Naskar[4]**
[1]Future Institute of Engineering and Management, India
[2]Saarland University, Germany,
[3]German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
[4]Jadavpur University, India
basuprasen@gmail.com, santanu.pal@uni-saarland.de,
sudip.naskar@jdvu.ac.in

## Abstract

The paper presents our participation in the WMT 2018 shared task on word level quality estimation (QE) of machine translated (MT) text, i.e., to predict whether a word in MT output for a given source context is correctly translated and hence should be retained in the post-edited translation (PE), or not. To perform the QE task, we measure the similarity of the source context of the target MT word with the context for which the word is retained in PE in the training data. This is achieved in two different ways, using *Bag-of-Words* (*BoW*) model and *Document-to-Vector* (*Doc2Vec*) model. In the *BoW* model, we compute the cosine similarity while in the *Doc2Vec* model we consider the Doc2Vec similarity. By applying the Kneedle algorithm on the F1-mult vs. similarity score plot, we derive the threshold based on which OK/BAD decisions are taken for the MT words. Experimental results revealed that the Doc2Vec model performs better than the BoW model on the word level QE task.

## 1 Introduction

Evaluating and estimating quality of a machine translation (MT) system without referring the actual translation is now one of the key research areas in MT domain (Blatz et al., 2004; Specia et al., 2009). In a machine translated document quality estimation can be performed at various granularities like word level, phrase level or sentence level (Specia et al., 2010, 2013). Scarton et al. (2016) produced their task in WMT16 in document level quality estimation with winning result in two different models (Bojar et al., 2016). One model used discourse features and SVR and another model employed word embedding feature and Gaussian Process for quality estimation. (Biçici, 2017) predicted translation performance with referential translation machines at word level, sentence level and at phrase level. (Blain et al., 2017) submitted task on bi-lexical word embedding in WMT17 QE shared task, which produced promising results in sentence level Quality Estimation. Some studies (Fiederer and OBrien, 2009; Koehn, 2009; DePalma and Kelly, 2011; Zampieri and Vela, 2014) show that the quality of MT output along with PE can produce better result than human editor in certain situations.

In our work we mainly focus on word level quality estimation. The distributional structure of words was first described by (Harris, 1954). (Turian et al., 2010) illustrated representations of words in semi-supervised learning. Bengio et al. (2003) proposed neural probabilistic language model by using a distributed representation of words. Collobert and Weston (2008), described how a convolutional neural network architecture could be used to make different language processing predictions, such as semantically similar words, etc. Mnih and Hinton (2008) proposed a fast hierarchical language model along with a feature based algorithm which automatically builds word trees from data. Mikolov et al. (2013b) proposed vector representation of words with the help of negative sampling (instead of softmax function) that improves both word vector quality and training speed. Their work showed prediction of a word from a context by adding two word vectors from the same context. (Mikolov et al., 2013a) proposed a novel approach to represent words as fixed length vectors, widely known as word2vec model and they reported state-of-the-art performance on word similarity task. (Le and Mikolov, 2014) extend their model to vector representation of a document known as Paragraph Vector model or commonly Document-to-Vector (Doc2Vec) model.

This paper reports our submission in the WMT 2018 Shared Task on Word-Level Quality Estima-

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 759–764
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64084

tion (QE task-2) on English–German (IT domain) SMT data. The proposed model has been developed in two ways - one using the standard Bag-of-Words model and another using the Doc2Vec model. The motivation behind the use of Doc2Vec model is to achieve more accurate semantic similarity compared to the simple cosine similarity on Bag-of-Words model. The Doc2Vec model captures semantic similarity which the Bag-of-Words model can not. Our word level error estimation is mainly based on Translation Error Rate (Snover et al., 2006) between MT and PE.

## 2 Proposed Approach

Our system highlights the retention of a word in MT translation and thus it helps human post-editors to increase their productivity with less effort. Our QE system is built over the Translation Error Rate (TER) (Snover et al., 2006) alignment between MT output and the corresponding PE output in the training data. TER alignment shows whether words from MT data (hypothesis in TER) will be continued, deleted or substituted with respect to the PE data (reference in TER). Based on the TER alignment, we build binary classification models that suggests $OK$ for continuation and $BAD$ for deletion or substitution.

Our QE system follows two models: Bag-of-Words Model and Document-to-Vector based model as described in the following subsections.

### 2.1 Bag-of-Words Model

MT words that are retained in PE are identified through TER alignment. In the Bag-of-words (BoW) model, for each word ($w_i$) in MT that is retained in PE in the training set, we find the corresponding source texts ($src^*_{w_i}$). A BoW ($B_{w_i}$) is then formed from the $src^*_{w_i}$ for each such $w_i$ that are present in both MT and the corresponding PE in the training set. Algorithm 1 presents the BoW creation method. $B_{w_i}$ contains more repetition of the source words which actually bear the meaning of $w_i$.

On the development set, we also establish TER alignment between the MT text ($MT_{dev}$) and the PE text ($PE_{dev}$). For each word (say, $w_j$) appearing in each sentence in $MT_{dev}$, we consider the corresponding $src$ as the source context (say, $src_{w_j}$) and keep track of the post-editing operation required on the word (through TER alignment), i.e., whether the word is retained ($OK$) in PE or not ($BAD$). Then we compute the cosine similarity between $src_{w_j}$ and $B_{w_j}$.

```
Input: $src$–$mt$–$pe$ parallel training data and TER
       alignments between $mt$ and $pe$
Output: source BoW ($B_{dict}$) for each target word
begin
    $V_{list} \leftarrow NULL$
    $B_{dict} \leftarrow NULL$
    foreach $sentence\ mt_i \in mt$ do
        foreach $T_{i,j} \in mt_i$ do
            if $T_{i,j}$ is retained in $pe_i$ then
                if $T_{i,j} \notin V_{list}$ then
                    $V_{list}.add(T_{i,j})$
                end
                $B_{list} \leftarrow NULL$
                forall $S_{i,k} \in src_i$ do
                    $B_{list}.add(S_{i,k})$
                end
                $B_{dict}[T_{i,j}].add(B_{list})$
            end
        end
    end
    return $B_{dict}$
end
```

**Algorithm 1:** Creation of source BoW; $T_{i,j}$ is the $j^{th}$ word of the $i^{th}$ $mt$ sentence and $S_{i,k}$ is the $k^{th}$ word of $i^{th}$ $src$ sentence.

The similarity scores range between 0 and 1 with varying distribution. We aim to arrive at a threshold on the similarity score above which the system takes the $OK$ decision, otherwise the $BAD$ decision. This threshold is trained on the development set. However, the datasets, both training and development, are highly imbalanced; 85.66% and 83% of the $mt$ tokens are retained (i.e., $OK$) in $pe$ in the training set and the development set respectively, and the rest are discarded or changed (i.e., $BAD$), which indicates that the $mt$ data was generated by a strong MT system. Such imbalance in the dataset proves to be a major hurdle in automatic QE or post-editing. The imbalance in the dataset leads to the fact that a very simple baseline of setting the threshold to 0 results in 85% F1-score on the development set (we consider only the non-stop words), which is very difficult to defeat.

The similarity scores obtained for the development set MT words are divided into a number of segments (or ranges) for equal distribution such that there are roughly equal number of instances in each range (cf. Table 3). The upper bound of each segment corresponds to a threshold.

We compute $F_1$-*mult*[1] for each of the segments

[1] $F_1$-*mult* is the multiplication of $F_1$ scores for the $OK$ and $BAD$ classes, and is the official evaluation metric for the WMT QE shared task.

and produce the $F_1$-*mult* curve. Figure 1 shows the $F_1$-*mult* curve on the development set which does not lead to any peak or intermediate threshold. We use the Kneedle algorithm (Satopaa et al., 2011) to find a knee point on the $F_1$-*mult* curve which serves as the threshold for our model and based on this threshold we take the $OK/BAD$ decision.

For each test set $MT$ word (say $w_k$), we generate the similarity score between $B_{w_k}$ and the current source $src_{w_k}$. If the score is above the threshold, the word is predicted as $OK$, otherwise $BAD$.

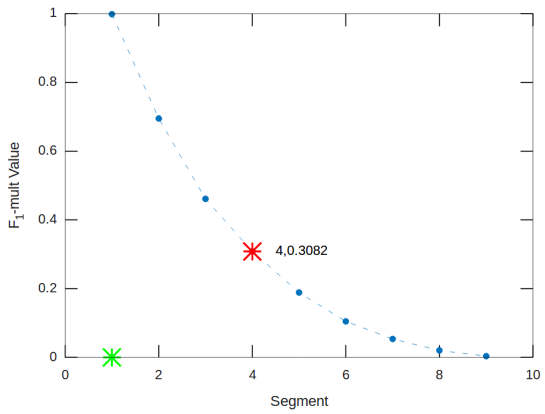


Figure 1: Segment vs. $F_1$-*mult* plot on the development set for the $BoW$ model. Red Mark denotes the (segment, $F_1$-*mult*) co-ordinate value for knee point and green Mark describes segment starting position.

### 2.2 Document-to-Vector based Model

In the Document-to-Vector (Doc2Vec) model for QE, for an MT word $w_i$, we also compute similarity between $src_{w_i}$ and $B_{w_i}$. However, here instead of the considering them as BoW, we treat them as documents and measure their Doc2Vec similarity score ($Sim_{D2V}$). For this, we prepare document vector for each $src_{w_i}$ and $B_{w_i}$ using gensim (Rehurek and Sojka, 2010). Gensim has its own implementation of document embedding via distributed memory or distributed Bag-of-Words model. In its model each document is represented as a fixed length vector. It is a generalization of and derived from the word2vec model. The QE decision is taken based on whether the $Sim_{D2V}$ for the word is above or below the threshold which is trained on the development set, as in the case of the BoW model. To train our Doc2Vec model we remove all stop words from the training data. For obtaining the threshold, the Doc2Vec similarity scores are divided into a number of segments of equal distribution. Like the BoW model, we generate the $F_1$-*mult* curve on those similarity scores and use the Kneedle algorithm to find the threshold.

## 3 Experiments

We used the WMT-2018 English–German (EN–DE) word level QE dataset for our experiments. Table 3 presents the statistics of the training, development and test sets. Stop words generally occur very frequently and their number of occurrences across BoW could easily mislead word-level QE. Therefore we process the training data by removing stop words for both German[2] and English from all the data sets, i.e., neither we consider them while building our context bags, nor we consider their QE.

| | Sentences | Tokens | | |
|---|---|---|---|---|
| | | *src* | *mt* | *pe* |
| Train | 26,299 | 389,070 | 393,000 | 400,058 |
| Dev | 1000 | 14,600 | 14,773 | 14,970 |
| Test | 1926 | 28,312 | 28,785 | - |

Table 1: Statistics of the the WMT-2018 Word Level QE Shared Task Data Set.

We considered 9 thresholds for the BoW model. Table 3 shows the segments and the corresponding thresholds.

| Seg. No | Threshold |
|---|---|
| 1 | 0.075 |
| 2 | 0.15 |
| 3 | 0.2 |
| 4 | 0.25 |
| 5 | 0.31 |
| 6 | 0.38 |
| 7 | 0.47 |
| 8 | 0.58 |
| 9 | 1 |

Table 2: Segment versus Threshold values for the BOW model

Table 3 shows word specific assignment of binary scores to each threshold. For a word with QE decision $OK$, a word–threshold cell is assigned to 1 if the similarity score for the corresponding word is higher than the corresponding threshold, and

[2] https://www.ranks.nl/stopwords/german

| Token | PE Decision | Score | Threshold | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0.075 | 0.15 | 0.2 | 0.25 | 0.31 | 0.38 | 0.47 | 0.58 | 1 |
| hinzugefgt | OK | 0.32 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| verhalten | OK | 0.26 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| zustzliche | BAD | 0.23 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| verknpfen | OK | 0.23 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| wird | OK | 0.32 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| verborgene | OK | 0.37 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| enthlt | BAD | 0.03 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| balken | OK | 0.21 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| fenster | OK | 0.17 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| sol | BAD | 0.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 3: A snapshot of the intermediate table showing word–threshold pair assignment

| Seg No | Th. Value |
|---|---|
| 1 | 0.001 |
| 2 | 0.12 |
| 3 | 0.21 |
| 4 | 0.4 |
| 5 | 0.1 |

Table 4: Segment vs. threshold values for the Doc2Vec model

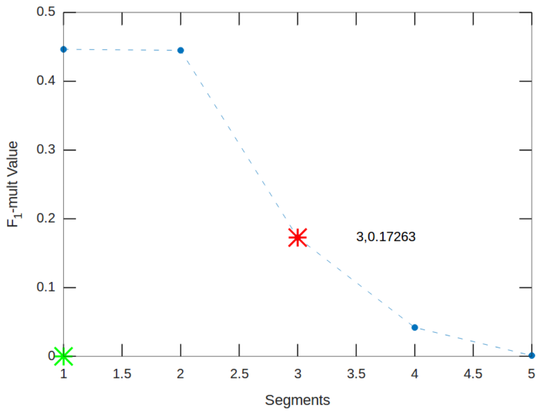


Figure 2: Segments versus $F_1$-$mult$ plot on training set of Doc2Vec model. Red Mark denotes the (segment, $F_1$-$mult$) co-ordinate value for knee point and green Mark describes segment starting position.

0 otherwise. For words with PE decision $BAD$, scores are assigned the other way round. It is to be noted that our model can only predict the QE decision for words that are already seen in the training set. Words that are not present in the training set (including stop words) are simply retained.

Kneedle algorithm on the Segments vs. $F_1$-$mult$ plot on the development set (cf. Figure 1) leads to the segment 4 as the knee point and the corresponding similarity score of 0.25 (cf. Table 3) serves as the threshold, which produces the optimal $F_1$-$mult$ for the BoW model.

For the Doc2Vec based experiment, gensim creates models using distributed Bag-of-Words. Doc2Vec similarity is measured between the vector representation of the Bag-of-Words and the source context for each target word from training data. The scores were distributed among 5 segments (cf. Table 3). Figure 2 shows the Segments vs. $F_1$-$mult$ plot for the Doc2Vec model. From the plot we take the knee value of the graph, i.e. segment 3 and the corresponding similarity score 0.21 (cf. Table 3) is considered as the threshold for the Doc2Vec model.

According to the WMT18 published results for the word level quality estimation task (Task 2), the results of our two models along with baseline are shown in Table 3. The evaluation results suggest that the Doc2Vec based word level QE model performs better than the Bag-of-Words based model for both the $OK$ class and the $BAD$ class on the WMT18 testset.

The expected results could have been better if we could use larger dataset as Doc2Vec model performs better for bigger data sources (Azunre et al., 2018). For Bag-of-Words based model we have removed stop words from those Bag-of-Words for the target German word of MT which itself is not a stop word. We also removed all stop words from test data. Removal of stop words from training data and test data leads to not-up-to-the-mark performance.

| *Participant* | *Model* | $F_1$-$BAD$ | $F_1$-$OK$ | $F_1$-$mult$ |
|---|---|---|---|---|
| fblain | BASELINE | 0.4115 | 0.8821 | 0.3630 |
| basuprasen | Doc2Vec | 0.2889 | 0.7547 | 0.2180 |
| basuprasen | BagOfWords | 0.2784 | 0.7335 | 0.2042 |

Table 5: Evaluation Results on the WMT18 Word level Quality Estimation (Task 2)

## 4 Conclusions and Future Work

The paper reports our participation in the WMT 2018 shared task on word level quality estimation (QE task2) on English–German SMT data. The task of word level QE is treated as a binary classification problem — i.e., decision is taken about whether a word under consideration is to be retained or not. The prediction is performed by measuring the similarity of the source context of the target word with the context for which the word is retained. This is achieved in two ways, using BoW model and Doc2Vec. Experimental results suggest that the Doc2Vec model can model this much more effectively than the Bag-of-Words model. An obvious extension of this work would be to extend our model to phrase-level QE and determining missing words and source words that lead to errors.

## Acknowledgments

Santanu Pal is partly funded by the German research foundation (DFG) under grant number GE 2819/2-1 (project MMPE). Sudip Kumar Naskar is supported by Digital India Corporation (formerly Media Lab Asia), MeitY, Government of India, under the Young Faculty Research Fellowship of the Visvesvaraya PhD Scheme for Electronics & IT. We also want to thank the reviewers for their valuable input, and the organizers of the shared task.

## References

Paul Azunre, Craig Corcoran, David Sullivan, Garrett Honke, Rebecca Ruppel, Sandeep Verma, and Jonathon Morgan. 2018. Abstractive tabular dataset summarization via knowledge basesemantic embeddings. *arXiv preprint arXiv:1804.01503*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Ergun Biçici. 2017. Predicting translation performance with referential translation machines. In *Proceedings of the Second Conference on Machine Translation*, pages 540–544.

Frédéric Blain, Carolina Scarton, and Lucia Specia. 2017. Bilexical embeddings for quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 545–550.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 315. Association for Computational Linguistics.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *ACL 2016 FIRST CONFERENCE ON MACHINE TRANSLATION (WMT16)*, pages 131–198. The Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Donald A. DePalma and Nataly Kelly. 2011. Project management for crowdsourced translation: How user-translated content projects work in real life. *Translation and Localization Project Management: The art of the possible*, XVI:379–408.

Rebecca Fiederer and Sharon OBrien. 2009. Quality and machine translation: A realistic objective. *The Journal of Specialised Translation*, 11:52–74.

Zellig S. Harris. 1954. Distributional structure. *WORD*, 10(2-3):146–162.

Philipp Koehn. 2009. A process study of computer-aided translation. *Machine Translation*, 23:241–263.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Andriy Mnih and Geoffrey Hinton. 2008. A scalable hierarchical distributed language model. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, pages 1081–1088, USA. Curran Associates Inc.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops*, ICDCSW '11, pages 166–171, Washington, DC, USA. IEEE Computer Society.

Carolina Scarton, Daniel Beck, Kashif Shah, Karin Sim Smith, and Lucia Specia. 2016. Word embeddings and discourse information for quality estimation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 831–837.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine translation*, 24(1):39–50.

Lucia Specia, Kashif Shah, Jose GC Souza, and Trevor Cohn. 2013. Quest-a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marcos Zampieri and Mihaela Vela. 2014. Quantifying the influence of mt output in the translators' performance: A case study in technical translation. In *EACL Workshop on Humans and Computer-assisted Translation (HaCat)*, pages 93–98.

# RTM results for Predicting Translation Performance

**Ergun Biçici**
ergun.bicici@boun.edu.tr
Department of Computer Engineering, Boğaziçi University
orcid.org/0000-0002-2293-2031
bicici.github.com

## Abstract

With improved prediction combination using weights based on their training performance and stacking and multilayer perceptrons to build deeper prediction models, RTMs become the 3rd system in general at the sentence-level prediction of translation scores and achieve the lowest RMSE in English to German NMT QET results. For the document-level task, we compare document-level RTM models with sentence-level RTM models obtained with the concatenation of document sentences and obtain similar results.

## 1 Introduction

Quality estimation task in WMT18 (Specia et al., 2018) (QET18) address machine translation performance prediction (MTPP), where translation quality is predicted without using reference translations, at the sentence- (Task 1), word- (Task 2), phrase-level (Task 3), and document-levels (Task 4). The tasks contain subtasks involving English-German phrase-based machine translation (SMT) and neural network-based SMT (NMT), German-English SMT, English-Latvian SMT and NMT, English-Czech SMT, and English-French SMT. Task 1 is about predicting HTER (human-targeted translation edit rate) scores (Snover et al., 2006), Task 2 is about binary classification of words, Task 3 is about binary classification of phrases, and Task 4 is about predicting multi-dimensional quality metrics (MQM) (Lommel, 2015).

We use referential translation machine (RTM) (Biçici, 2017) models for building our prediction models. RTMs predict data translation between the instances in the training set and the test set using interpretants, data close to the task instances. Interpretants provide context for the prediction task and are used during the derivation of the features measuring the closeness of the test sentences to the training data, the difficulty of translating them, and to identify translation acts between any two data sets for building prediction models. With the enlarging parallel and monolingual corpora made available by WMT, the capability of the interpretant datasets selected by RTM models to provide context for the training and test sets improve.

|  |  |  | RTM interpretants |  |
|---|---|---|---|---|
| Task | Train | Test | Training | LM |
| Task 1 (en-cs, SMT) | 41254 | 1000 | 0.225M | 5M |
| Task 1 (en-de, SMT) | 27273 | 1000 |  |  |
| Task 1 (en-de, NMT) | 14442 | 1000 |  |  |
| Task 1 (de-en, SMT) | 26963 | 1000 |  |  |
| Task 1 (en-lv, SMT) | 12251 | 1000 |  |  |
| Task 1 (en-lv, NMT) | 13936 | 1000 |  |  |
| Task 1 (en-lv, NMT) | 13936 | 1000 |  |  |
| Task 3 (de-en, NMT) | 6021 | 543 |  |  |
| Task 4 (en-fr, NMT) | 1200 | 269 |  |  |

Table 1: Number of instances and interpretants used.

Figure 1 depicts RTMs and explains the model building process. RTMs use parfda (Bicici, 2018) for instance selection and machine translation performance prediction system (MTPPS) (Biçici and Way, 2015) for generating features. The total number of features vary depending on the order of $n$-grams used (e.g. a log of probability score from the language model for each $n$-gram is used).

We use ridge regression, kernel ridge regression, k-nearest neighbors, support vector regression, AdaBoost (Freund and Schapire, 1997), gradient tree boosting, extremely randomized trees (Geurts et al., 2006), and multi-layer perceptron (Bishop, 2006) as learning models in combination with feature selection (FS) (Guyon et al., 2002) and partial least squares (PLS) (Wold et al., 1984) where most of these models can be found in scikit-learn. [1] Evaluation metrics listed

[1] http://scikit-learn.org/

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 765–769
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64085

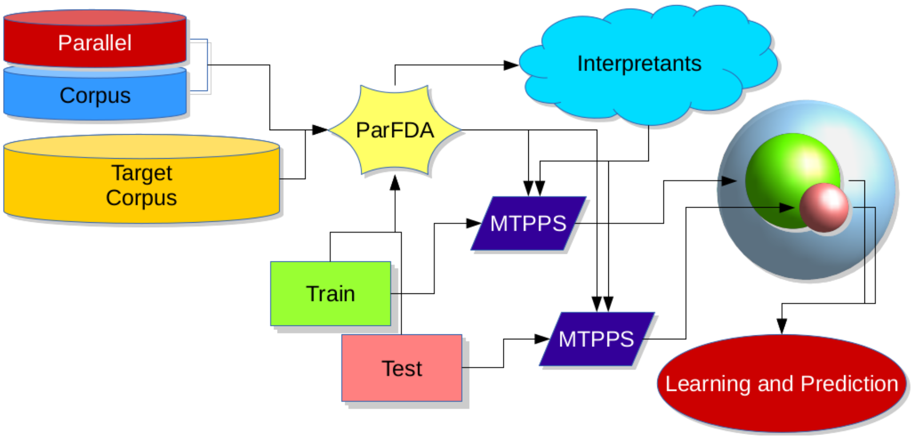


Figure 1: RTM depiction: ParFDA selects interpretants close to the training and test data using parallel corpus in bilingual settings and monolingual corpus in the target language or just the monolingual target corpus in monolingual settings; an MTPPS use interpretants and training data to generate training features and another use interpretants and test data to generate test features in the same feature space; learning and prediction takes place taking these features as input.

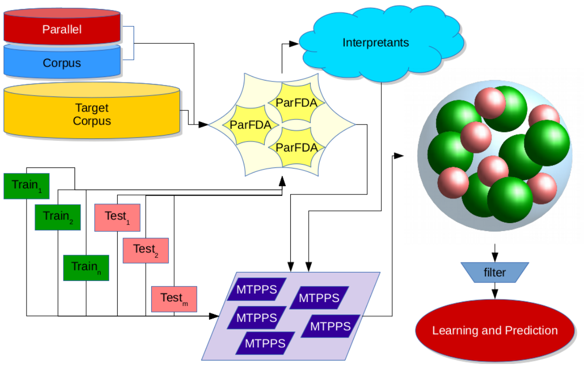


Figure 2: Document-level RTM model with separate MTPPS run for each training and test document to obtain corresponding feature representations, which are filtered and processed before learning and prediction.

are Pearson's correlation ($r$), mean absolute error (MAE), and root mean squared error (RMSE).

We use Global Linear Models (GLM) (Collins, 2002) with dynamic learning (GLMd) (Biçici, 2017) for word- and phrase-level translation performance prediction. GLMd uses weights in a range $[a, b]$ to update the learning rate dynamically according to the error rate.

## 2 Mixture of Experts Models

We use prediction averaging (Biçici, 2017) to obtain a combined prediction from various prediction outputs better than the components, where the performance on the training set is used to obtain weighted average of the top $k$ predictions, $\hat{y}$ with evaluation metrics indexed by $j \in J$ and weights

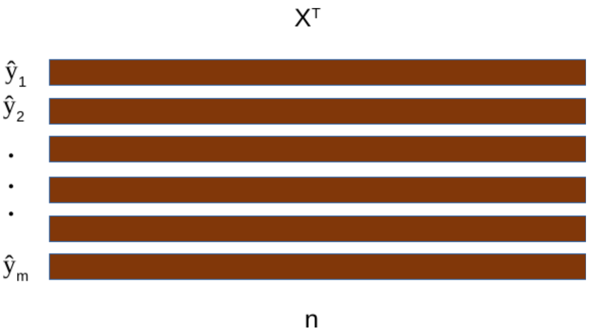


Figure 3: Stacking training data, X, from $m$ predictors.

with $w$:

$$\begin{aligned}
w_{j,i} &= \begin{cases} \frac{1}{\text{eval}_{j,i}} & \text{if } j \text{ is minimized} \\ \text{eval}_{j,i} & \text{if } j \text{ is maximized} \end{cases} \\
\hat{\boldsymbol{y}}_{\mu_k} &= \tfrac{1}{k} \textstyle\sum_{i=1}^{k} \hat{\boldsymbol{y}}_i && \text{MEAN} \\
\hat{\boldsymbol{y}}_{j,w_k^j} &= \frac{1}{\sum_{i=1}^{k} w_{j,i}} \textstyle\sum_{i=1}^{k} w_{j,i}\, \hat{\boldsymbol{y}}_i \\
\hat{\boldsymbol{y}}_k &= \frac{1}{|J|} \textstyle\sum_{j \in J} \hat{\boldsymbol{y}}_{j,w_k^j} && \text{MIX}
\end{aligned} \tag{1}$$

where weights are inverted to decrease error. We only use the MIX prediction if we obtain better results on the training set. We select the best model using $r$ and mix the results using $r$, RAE, MRAER, and MAER. The set of evaluation metrics used for mixing also affects the results. Since we try to obtain results with relative evaluation metric scores less than 1, we filter out those results with higher than 1 relative evaluation metric scores.

In our experiments, we found that assuming independent predictions and using $p_i/(1 - p_i)$ for

| Task 4 | model | setting | $r_P$ | MAE | RAE | MAER | MRAER |
|---|---|---|---|---|---|---|---|
| en-fr SMT | doc | stack | -0.1725 | 43.0687 | 0.9857 | 0.8123 | 0.805 |
| en-fr SMT | doc | mix | -0.1812 | 43.5726 | 0.9973 | 0.8347 | 0.8256 |
| en-fr SMT | doc | FS RR | -0.177 | 44.058 | 1.0084 | 0.8509 | 0.8413 |
| en-fr SMT | sent | stack | 0.2109 | 42.5196 | 0.9732 | 0.8464 | 0.8372 |
| en-fr SMT | sent | mix | -0.2299 | 43.2402 | 0.9897 | 0.8197 | 0.8116 |
| en-fr SMT | sent | FS KR | -0.1844 | 43.2891 | 0.9908 | 0.8255 | 0.8167 |

Table 2: Training results on Task 4 with stacking and prediction averaging. FS RR is the top single model for doc and FS KR for sent where RR is ridge regression and KR is kernel ridge regression.

| Task 1 | $r_P$ | MAE | RAE | MAER | MRAER |
|---|---|---|---|---|---|
| en-de SMT | 0.4336 | 0.1365 | 0.8654 | 0.7951 | 0.798 |
| en-de NMT | 0.459 | 0.1138 | 0.8282 | 0.84 | 0.7771 |
| de-en SMT | 0.5634 | 0.1364 | 0.7953 | 0.7637 | 0.7573 |
| en-cs SMT | 0.5381 | 0.151 | 0.8109 | 0.7423 | 0.7679 |
| en-lv SMT | 0.3805 | 0.1362 | 0.9055 | 0.8755 | 0.8041 |
| en-lv NMT | 0.5714 | 0.1466 | 0.7971 | 0.753 | 0.7595 |

Table 3: Training results on Task 1 with prediction averaging.

weights where $p_i$ represents the accuracy of the independent classifier $i$ in a weighted majority ensemble (Kuncheva and Rodríguez, 2014) obtained slightly better results (Equation (2)).

$$w_{j,i} = \frac{w_{j,i}}{1 - w_{j,i}} \quad (2)$$

We also use stacking to build higher level models using predictions from base prediction models where they can also use the probability associated with the predictions (Ting and Witten, 1999). The stacking models use the predictions from predictors as features and build second level predictors (Figure 3).

## 3 Document-level MTPP Model Comparisons

We evaluate the effect of two different RTM data modeling techniques for the document-level task. Our first approach involves running separate MTPPS instances for each training (green in Figure 2) or test (salmon colored) document to obtain specific features for each document. Then, only the document-level features and the min, max, and average of the sentence-level features are used to obtain an RTM representation vector instance from each document. Our second approach concatenates the sentences from each document to obtain a single sentence representing each and runs an RTM model. Features from word alignment are included in both and they share the interpretants. The first approach use 1359 features and the second use 383 features.

| Task | | Model | % error |
|---|---|---|---|
| Task 2 | word | en-de SMT | 0.080 |
| | | en-de NMT | 0.032 |
| | | de-en SMT | 0.066 |
| | | en-cs SMT | 0.116 |
| | | en-lv SMT | 0.027 |
| | | en-lv NMT | 0.058 |
| | gap | en-de SMT | |
| | | en-de NMT | 0.017 |
| | | de-en SMT | 0.040 |
| | | en-cs SMT | |
| | | en-lv SMT | 0.030 |
| | | en-lv NMT | 0.017 |
| Task 3 | word | de-en SMT | 0.020 |
| | phrase | | 0.015 |
| | word gap | | 0.030 |
| | phrase gap | | 0.011 |

Table 4: RTM Task 2 training error for some of the models where GLMd is parallelized over splits. All GLMd models use $[0.5, 2]$ as weights. $\%$ error are twice the overall error found based on all tags (2N+1).

Training results are in Table 2 where we compare them and the first approach is denoted as doc and the second as sent. The first approach obtained the top results in QET16 (Bicici, 2016). doc obtains better MAER (mean absolute error relative) and MRAER (mean relative absolute error relative) (Biçici and Way, 2015). We obtain 3rd best RMSE while we note that both MAE and RMSE results are close to each other in all four submissions on the test set.

## 4 Results

Table 1 lists the number of sentences in the training and test sets for each task and the number of instances used as interpretants in the RTM models (M for million). We tokenize and truecase all of the corpora using Moses' (Koehn et al.,

|  | Task 4 | model | setting | $r_P$ | MAE | RMSE |
|---|---|---|---|---|---|---|
| top |  |  |  | 0.5337 | 56.2264 | 85.2319 |
| RTM | en-fr SMT | doc | stack | 0.0580 (4) | 58.5680 (4) | 87.8321 (4) |
|  | en-fr SMT | doc | mix | -0.1210 (4) | 57.5613 (4) | 86.2219 (4) |
|  | en-fr SMT | sent | stack | 0.0183 (4) | 57.6245 (4) | 86.4831 (4) |
|  | en-fr SMT | sent | mix | -0.0812 (4) | 57.7922 (4) | 86.8650 (4) |

Table 5: Task 4 test RTM results and the top result in the task.

| Task 1 |  | $r_P$ | $r_S$ | MAE | RMSE |
|---|---|---|---|---|---|
| en-de SMT | top | 0.7397 | 0.7543 | 0.0937 | 0.1362 |
|  | RTM | 0.4166 (6) | 0.4254 (4) | 0.1353 (5) | 0.1731 (6) |
| en-de NMT | top | 0.5129 | 0.6052 | 0.1114 | 0.1719 |
|  | RTM | 0.4704 (3) | 0.5461 (3) | 0.1192 (3) | **0.1727 (1)** |
| de-en SMT | top | 0.7667 | 0.7318 | 0.0945 | 0.1315 |
|  | RTM | 0.5772 (6) | 0.5167 (5) | 0.1311 (6) | 0.1679 (4) |
| en-cs SMT | top | 0.6918 | 0.7105 | 0.1223 | 0.1693 |
|  | RTM | 0.5295 (3) | 0.5348 (3) | 0.1519 (3) | 0.1952 (3) |
| en-lv SMT | top | 0.6188 | 0.5766 | 0.1202 | 0.1602 |
|  | RTM | 0.3521 (8) | 0.2861 (7) | 0.1430 (4) | 0.1869 (3) |
| en-lv NMT | top | 0.6819 | 0.6665 | 0.1308 | 0.1747 |
|  | RTM | 0.5487 (4) | 0.5017 (4) | 0.1540 (3) | 0.2006 (3) |

Table 6: Test results of RTM in Task 1 where numbers in parentheses show the rank and corresponding top results. RTM achieves the lowest RMSE in en-de NMT and becomes the 3rd system in general. $r_P$ is Pearson's correlation and $r_S$ is Spearman's correlation.

2007) processing tools. [2] LMs are built using `kenlm` (Heafield et al., 2013). The comparison of results on the training set are in Table 3 for Task 1 and in Table 2 for Task 4.

The results on the test set (Tables 5 and 6) shows that RTM can become the 1st in en-de NMT and 3rd in general. Test results are taken from the competition's result submission websites at:

- sentence level `https://competitions.codalab.org/competitions/19316`
- word level `https://competitions.codalab.org/competitions/19306`
- phrase level `https://competitions.codalab.org/competitions/19308`
- document level `https://competitions.codalab.org/competitions/19309`

The references for the test sets are not released yet.

For Task 2 and Task 3, we model words or phrases and gaps separately and then combine their results. The error % on the training sets are in Table 4 and the results on the test set are in Table 7.

|  | Model | task | $F_1$ BAD | $F_1$ OK | w$F_1$ |
|---|---|---|---|---|---|
| word | en-de SMT | word | 0.3300 (7) | 0.8813 (3) | 0.2908 (6) |
|  |  | gap | 0.2547 (3) | 0.9764 (1) | 0.2487 (3) |
|  |  | src | 0.1650 (2) | 0.8591 (1) | 0.1418 (2) |
|  | en-de NMT | word | 0.0927 (6) | 0.9235 (1) | 0.0856 (6) |
|  |  | gap | 0.1360 (1) | 0.9878 (1) | 0.1343 (1) |
|  |  | src | 0.0337 (2) | 0.9209 (1) | 0.0310 (2) |
|  | de-en SMT | word | 0.3790 (6) | 0.8979 (3) | 0.3403 (6) |
|  |  | gap | 0.1463 (3) | 0.9804 (1) | 0.1435 (3) |
|  |  | src | 0.1211 (2) | 0.8946 (1) | 0.1083 (2) |
|  | en-lv SMT | word | 0.3681 (3) | 0.9044 (1) | 0.3329 (3) |
|  |  | gap | 0.1298 (3) | 0.9853 (1) | 0.1279 (3) |
|  |  | src | 0.1195 (2) | 0.8917 (1) | 0.1066 (2) |
|  | en-lv NMT | word | 0.4280 (4) | 0.8530 (1) | 0.3651 (3) |
|  |  | gap | 0.0829 (3) | 0.9819 (1) | 0.0814 (3) |
|  |  | src | 0.1977 (2) | 0.8418 (1) | 0.1664 (2) |
|  | en-cs SMT | word | 0.5280 (4) | 0.8257 (2) | 0.4360 (4) |
|  |  | gap | 0.1059 (3) | 0.9810 (1) | 0.1039 (3) |
|  |  | src | 0.3229 (2) | 0.7962 (2) | 0.2571 (2) |
| phrase | de-en SMT | phrase | 0.2651 (3) | 0.9168 (1) | 0.2431 (2) |
|  |  | gap | 0.0518 (2) | 0.9811 (1) | 0.0508 (2) |
|  |  | src | 0.0956 (1) | 0.8994 (1) | 0.0860 (1) |
|  |  | word | 0.1648 (3) | 0.9004 (2) | 0.1484 (3) |
|  |  | gap | 0.1029 (2) | 0.9373 (1) | 0.0964 (2) |
|  |  | src | 0.0973 (2) | 0.8376 (1) | 0.0815 (2) |

Table 7: RTM Task 2 and Task 3 results on the test set. w$F_1$ is average weighted $F_1$ score ($F_1$ multi).

## 5 Conclusion

Referential translation machines can achieve top performance in automatic, accurate, and language independent prediction of translation scores and achieve to become the 1st system according to RMSE for MTPP from English to German in QET18. RTMs pioneer a language independent approach and remove the need to access any task or domain specific information or resource.

## Acknowledgments

The research reported here received financial support in part from the Scientific and Technological Research Council of Turkey (TÜBİTAK) with-

[2] `https://github.com/moses-smt/mosesdecoder/tree/master/scripts`

out contribution to the content nor responsibility thereof. We also thank the reviewers' comments and Fred Blain from The University of Sheffield.

## References

Ergun Biçici. 2017. Predicting translation performance with referential translation machines. In *Proc. of the Second Conference on Statistical Machine Translation (WMT17)*, pages 540–544, Copenhagen, Denmark. Association for Computational Linguistics.

Ergun Biçici and Andy Way. 2015. Referential translation machines for predicting semantic similarity. *Language Resources and Evaluation*, pages 1–27.

Ergun Bicici. 2016. Referential translation machines for predicting translation performance. In *Proc. of the First Conference on Statistical Machine Translation (WMT16)*, pages 777–781, Berlin, Germany. Association for Computational Linguistics.

Ergun Bicici. 2018. Robust parfda statistical machine translation results. In *Proc. of the Third Conference on Statistical Machine Translation (WMT18)*, Brussels, Belgium.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning.*

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proc. of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 1–8, Stroudsburg, PA, USA.

Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria.

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. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proc. of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics.

Ludmila I. Kuncheva and Juan J. Rodríguez. 2014. A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems*, 38(2):259–275.

Arle Lommel. 2015. Multidimensional quality metrics (mqm) definition. *URL http://www. qt21. eu/mqm-definition/definition-2015-12-30. html.*

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proc. of Association for Machine Translation in the Americas,.*

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón F. Astudillo, and André Martins. 2018. Findings of the wmt 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Kai Ming Ting and Ian H. Witten. 1999. Issues in stacked generalization. *Journal of Artificial Intelligence Research*, 10:271–289.

S. Wold, A. Ruhe, H. Wold, and III Dunn, W. J. 1984. The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5:735–743.

# Neural Machine Translation for English-Tamil

**Himanshu Choudhary**
DTU-Delhi
himanshu.dce12@gmail.com

**Aditya Kumar Pathak**
IIIT-Bhubaneswar
adityapathak.cse@gmail.com

**Rajiv Ratn Shah**
IIIT-Delhi
rajivratn@iiitd.ac.in

**Ponnurangam Kumaraguru**
IIIT-Delhi
pk@iiitd.ac.in

## Abstract

A huge amount of valuable resources is available on the web in English, which are often translated into local languages to facilitate knowledge sharing among local people who are not much familiar with English. However, translating such content manually is very tedious, costly, and time-consuming process. To this end, machine translation is an efficient approach to translate text without any human involvement. Neural machine translation (NMT) is one of the most recent and effective translation technique amongst all existing machine translation systems. In this paper, we apply NMT for English-Tamil language pair. We propose a novel neural machine translation technique using word-embedding along with Byte-Pair-Encoding (BPE) to develop an efficient translation system that overcomes the OOV (Out Of Vocabulary) problem for languages which do not have much translations available online. We use the BLEU score for evaluating the system performance. Experimental results confirm that our proposed MIDAS translator (8.33 BLEU score) outperforms Google translator (3.75 BLEU score).

## 1 Introduction

Big countries such as India and China have several languages which change by regions. For instance, India has 23 constitutionally recognized official languages (*e.g.*, Hindi, Tamil, and Panjabi) and several hundreds unofficial local languages. Despite Indian population is approximately 1.3 billion, only approximately 10% of them English speak English. Some studies say that out of these 10% English speakers only 2% can speak, write, and read English well, and rest 8% can merely understand simple English and speak broken English with an amazing variety of accents (sta). Considering a significant amount of valuable resources is available on the web in English and most people in India can not understand it well, it is essential to translate such content in to local languages to facilitate people. Sharing information between people is necessary not only for business purposes but also for sharing their feelings, opinions, and acts. To this end, translation plays an important role in minimizing the communication gap between different people. Considering the vast amount of information, it is not feasible to translate the content manually. Hence, it is essential to translate text from one language (say, English) to another language (say, Tamil) automatically. This process is also known as *machine translation*.

There are many challenges in machine translation for Indian languages. For instance, (i) the size of parallel corpora and (ii) differences amongst languages, mainly the morphological richness and word order differences due to syntactical divergence are two of the major challenges. Indian languages (IL) suffer both of these problems, especially when they are being translated from English. There are only a few parallel corpora for English and Indian languages. Moreover, Indian languages such as Tamil differ from English in word order as well as in morphological complexity. For instance, English has Subject-Verb-Object (SVO) whereas Tamil has Subject-Object-Verb (SOV). Moreover, English is a fusional whereas Tamil is agglutinative languages. While syntactic differences contribute to difficulties of translation models, morphological differences contribute to data sparsity. We attempt to address both issues in this paper.

Though much work is being done on machine translation for foreign and Indian languages but apart from foreign languages most of works on Indian languages are limited to conventional machine translation techniques. We observe that the techniques like word-embedding and Byte-pair-encoding (BPE) are not applied on many Indian languages which have shown a great improvement

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 770–775
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64086

in natural language processing. Thus, in this paper, we apply a neural machine translation technique (torch implementation) with word embedding and BPE. Especially, we work on English-Tamil language pair as it is one of the most difficult language pair (ZdenekŽabokrtskỳ, 2012) to translate due to morphologically richness of Tamil language. We obtain the data from EnTamv2.0 and Opus, and evaluate our result using widely used evaluation matric BLEU. Experimental results confirm that we got much better results than conventional machine translation techniques on Tamil language. We believe that our work can also be applied to other Indian language pairs too.

Main contributions of our work are as follows:

- This is the first work to apply BPE with word embedding on Indian language pair (English-Tamil) with NMT technique.
- We achieve comparable accuracy with a simpler model in less training time rather then training on deep and complex neural network which requires much time to train.
- We have shown how and why data preprocessing is a crucial step in neural machine translation.
- Our model outperforms Google translator with margin of 4.58 BLEU score.

The rest of the paper is organized as follows. Sections 2 and 3 describe related work and the methodology of our MIDAS translator, respectively. Evaluation is presented in Section 4. Finally, Section 5 concludes the paper.

## 2 Literature Survey

Several works have been reported on machine translation (MT) in last a few decades, earliest one in 1950s (Booth, 1955). There are various approaches adopted by researchers such as rule-based MT (Ghosh et al., 2014; Wong et al., 2006), corpus-based MT (Wong et al., 2006), and hybrid-based MT (Salunkhe et al., 2016). Each of these approaches has its own pros and cons. Rule-based machine translation systems traverse the source text to produce an intermediate representation of the text, and depending on the representation this approach is further classified into transfer-based approach (TBA)(Shilon, 2011) and inter-lingua based approach (IBA).[1]

Corpus-based approach uses a large sized parallel corpora in the form of raw data. This raw data contains text with their respective translations. These corpora are used to acquire knowledge for translation. A corpus-based approach divides itself into two sub types: (i) statistical machine translation (SMT) and (ii) example-based machine translation (EBMT) (Somers, 2003). SMT[2] generates its translation on the basis of statistical models. It depends on the combination of language model as well as translation model with a decoding algorithm. EBMT on the other hand uses the existing translation examples for generating a new translation. This is done by finding out the examples matching with the input. Then alignment is performed to find out the parts of translation that can be reused. Hybrid-base machine translation is a combination of transfer approach and any corpus-based approaches in order to overcome their limitations.

Recent research (Khan et al., 2017) suggest that the machine translation performance of Indian language pairs (*e.g.*, Hindi, Bengali, Tamil, Punjabi, Gujarati, and Urdu) is of an average of 10% accuracy. This necessitates the need of building better machine translation systems for Indian languages.

NMT is novel and emerging technique for various languages and shown remarkable results (Hans and Milton, 2016). In this paper phrase-based hierarchical models trained after morphological preprocessing using NMT. Patel *et al.* (Patel et al., 2017) trained their model after suffix separation and compound splitting. Different models were also tried for the same task and achieved a good result on their respective dataset (Pathak and Pakray). We analyze that morphological preprocessing, suffix separation, and compound splitting can be overpass by using Byte-Pair-Encoding and produced similar or better translation without making the model complex.

## 3 Methodology

In this study, we present a neural machine translation technique using word-embedding along with Byte-Pair-Encoding (BPE) to develop an efficient translation system, called MIDAS translator that

[1] https://en.wikipedia.org/wiki/Interlingual_machine_translation

[2] https://books.google.ch/books?id=4v_Cx1wIMLkC

overcomes the OOV (Out Of Vocabulary) problem for languages which do not have much translations available online. Thus, first, we provide an overview of neural machine translation, attention model, word embedding, and Byte Pair Encoding. Next, we present the framework of our MIDAS translator.

### 3.1 Neural Machine Translation Overview

Neural Machine translation is a technique that is based on neural networks and the conditional probability of translated sentence from the source language to target sentences (Revanuru et al., 2017). In the following sub-sections we will provide an overview of sequence to sequence architecture and attention model that are used in our proposed MIDAS translator.

**Sequence to Sequence Architecture** Sequence to sequence architecture is basically used for response generation whereas in machine translation models it is used to find the relationship between two different language pairs. It consists of two parts, an encoder and a decoder. The encoder takes the input from source and the decoder generates the output based on encoding vector and previously generated words. Assume $A$ be the source sentence and $B$ be a target sentence. The encoder converts the source sentence $a_1, a_2, a_3..., a_n$ into vector of fixed dimensions and the decoder outputs word by word using conditional probability. Here, $A_1, A_2, ..., A_M$ in the equation are the fixed size encoded vectors. Using chain rule, the Eq. 1 is converted to the Eq. 2.

$$P(B/A) = P(B|A_1, A_2, A_3, ..., A_M) \tag{1}$$

$$\begin{split} P(B|A) = P(b_i|b_0, b_1, b_2, ..., b_{i-1}; \\ a_1, a_2, a_3, ..., a_m \end{split} \tag{2}$$

While decoding, next word is predicted using previously predicted word vectors and source sentence vectors in Eq. 1. Each term in the distribution is represented with a softmax over all the words in the vocabulary.

**Attention Model** In a basic encoder-decoder architecture, encoder reads the whole sentence, memorizes it and store it in the final activation layer, then the decoder network generates the target translation. This architecture works quite well for short sentences, so we might achieve a relatively high BLEU score, but for very long sentences, maybe longer than 30 or 40 words, the performance degrades. Using attention[3] mechanism with a basic encoder-decoder architecture is a solution for that. It translates similar to humans by looking at part of the sentence at a time. The mechanism decides how much attention should be paid to a particular word while translating the sentence. The mechanism is shown in Fig. 2. The Encoder generates the attention vectors $h_1, h_2, h_3......h_t$ from the inputs $A_1, A_2, A_3 A_t$. Then, context vector $C_i$ is calculated using concatenation of these vector for each output time step. Then Using the context vector $C_i$ hidden state $S_i$ and previously predicted words, decoder generates the softmax output $B_i$.

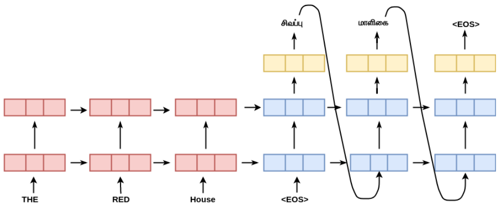


Figure 1: Seq2Seq architecture for English-Tamil

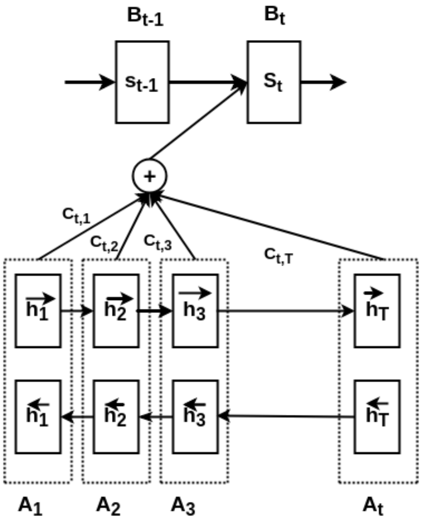


Figure 2: Attention model

**Word Embedding** Word embedding is a way of representing words on a vector space where the words having same meaning have similar vector representations. Each word from vocabulary is represented in hundreds of dimensions. Normally

[3] https://hackernoon.com/attention-mechanism

pre-trained word embeddings are used and with the help of transfer learning words from vocabulary are converted to vector (Cho et al., 2014). In our model, we used FastText word vectors[4] to convert English and Tamil vocabulary into a 300-dimensional vector. Training the model with same layers, optimization method, attention, and regularization we got a BLEU score of Point 6.74.

**Byte Pair Encoding** BPE (Gage, 1994) is a simple data compression technique. It replaces most frequent pair bytes in a sequence with single unused byte. We use this algorithm for word segmentation. By merging frequent pairs of bytes we merge charters or character sequences (Sennrich et al., 2015). NMT symbols interpretative as sub-words units and networks can translate and make the new word on the basis of sub-words. We learned the independent encodings on our source and target training data with 10,000 and 25,000 words and then apply it on train test and validation data for both source and target. BPE helped in compound splitting and suffix, prefix separation which is used for creating new words of Tamil language. we used BPE along with word embeddings and tried different models.

### 3.2 MIDAS Translator

We tried various models to get a better intuition on how parameter tuning along with different techniques affects on Indian language pair. Our first model architecture consists of 2 layer Bi-directional LSTM encoder and 2 layers LSTM decoder of 500 dimensions each with the vocabulary size of 50,004 words for both source and target. First we tried SGD optimization method, Luong attention with a dropout (regularization) of 0.3, and learning rate 1.0. Secondly, we changed the optimization method to Adam and attention to Bahdanau with the learning rate of 0.001. We got our best results with a BPE vocabulary size of 25,000 with 2 Layer Bi-directional encoder-decoder, Adam optimization with a learning rate of 0.001, Bahdanau attention, and word-embedding with the dimension of 500. We used GPU (Nvidia GeForce GTX 1080) for the training of different models which increase the computation speed. We achieved our result after 5 hours of training on this GPU.

## 4 Evaluation

### 4.1 Evaluation Metric

The BLEU score or bilingual evaluation under study is a method to measure the difference between machine and human translations (Papineni et al., 2002). The approach works by counting and matching n-grams in result translation to n-grams in the reference text, where unigram would be each token and a bigram comparison would be each word pair and so on. The comparison is made regardless of word order. This method is a modification of a simple precision method.

### 4.2 Dataset

We used the datasets obtained from EnTam V2.0[5] and Opus.[6] The sentences are taken from various domains like news, bible, cinema, movie subtitles and combined to build our final parallel dataset. After preprocessing and splitting it to train, test, and validation, our final dataset contains 1,83,451 training corpus, 1,000 validation and 2,000 test corpus from English to Tamil. The data used is encoded in UTF-8 format.

### 4.3 Data Pre-processing

Research works (Hans and Milton, 2016; Ramesh and Sankaranarayanan, 2018) suggest that they have used EnTamV2.0 in their experiments. However, we find that in both well-known parallel corpus for English-Tamil datasets (*i.e.*, EnTam V2.0 and Opus) have many repeated sentences, which outcomes the wrong results (may be high or low) after dividing into train, test, and validation sets, as some of the sentences occur both in train and test sets. Thus, it is essential to clean, analyses, and correct before using for experiment. We observed the following four main problems in the online available corpus for English-Tamil dataset.

- Repetition of sentences with same source and same target .
- Sentences with same source and different translation.
- Sentences with different source and same translation.
- Tokenization of Indian languages.

[4] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

[5] http://ufal.mff.cuni.cz/~ramasamy/parallel/html/

[6] http://opus.nlpl.eu/

To overcome the first problem we took unique pairs from all sentences and removed repeating ones. We completely removed those sentences which are repeated more than once because in the second case we cannot identify that which translated sentence is correct for the same source and which source is correct for the same translation in the third case. We observed that there are some sentences which are repeating even more than 10 times in Opus dataset. This confuses the model to learn and identify different new features, overfits the model, and led to the wrong results. This preprocessing is required as it may be possible that train and test contain the same sentences which let to the better prediction for test set but wrong predictions for new sentences.

The second important thing which we observed that there are many tools available for tokenization of English language (*e.g.*, Perl tokenizer) but does not work well for the Tamil language, because there are different morphological symbols which used in word formation of Tamil language which are removed by these tokenization tools in Indian languages (Tamil in our case). Without tokenization model consider *word*, *word,* and *word!* as three different words in the vocabulary of Tamil. We tokenize the Tamil language sentences using our own code before training. This problem can also be overcome by Byte-pair-Encoding.

Finally after working on all these small but effective preprocessing such as removing sentences with the length greater than 50, removing non translated words in target sentences, removing noisy translations and unwanted punctuations, we got our final dataset[7] of 1,86,451 parallel sentences which was cleared from 2,23,685 sentences. It is divided into training (1,83,451 sentences) testing (2,000 sentences) and validation (1,000 sentences) respectively after shuffling.

### 4.4 Result

We used Google translate API in python to translate the English sentences and compared Google results with our various models. It is also observed that the translations below are handy enough to use in day to day life as well as official works. From test results, we can also deduce that our model overcomes the OOV (Out of Vocabulary) problem in some cases.

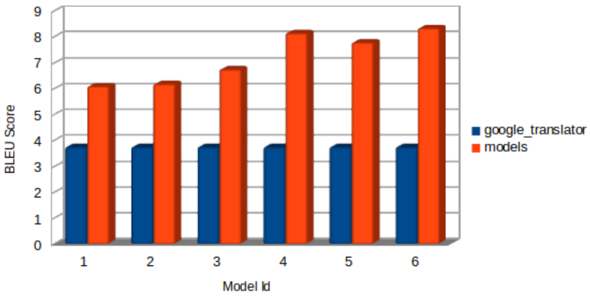


Figure 3: Different model comparison with Google Translator Table1.

| Model | BLEU |
|---|---|
| Google Translator | 3.75 |
| Bi-L+S+Lu | 6.10 |
| Bi-L+A+B | 6.18 |
| Bi-L+A+B+E | 6.74 |
| Bi-L(4-Layer)+A+B +BPE(10000)+E | 7.78 |
| Bi-L+A+B +BPE(10000)+E | 8.14 |
| Bi-L+A+B+BPE(25000)+E | 8.33 |

Table 1: BLEU Score of English-Tamil translated system. Symbols have the following meanings: Bi-L= Bi-LSTM, S= SGD(Wu et al., 2016), L= LSTM, A=Adam(Vaswani et al., 2017), B= Bahdanau (Bahdanau et al., 2014), E=Word Embedding, Lu=Luong(Luong et al., 2015))

### 4.5 Analysis

We conducted an anonymous survey of ten random sentences from test data and accumulated reviews of Tamil speaking people on that. After comparing accumulated reviews of Google translator and MIDAS translator, it was discovered that translations from our MIDAS translator are selected as better translations in 71.66% cases than translations of Google translator. Moreover, two out of ten translations from MIDAS translator are unanimously selected by respondents in compared to only one translation by Google translator.

## 5 Conclusion & Future Work

In this paper, we applied NMT to one of the most difficult language pairs (English-Tamil). We showed that NMT with pre-trained word embedding and Byte Pair Encoding performs better than complex translation techniques on Indian languages. Our model outperformed Google translator with a margin of 4.58 BLEU points. Since We achieved fairly good accuracy so our model can

[7] `https://github.com/himanshudce/MIDAS-NMT-English-Tamil`

be used for creating English-Tamil translation applications that will be useful in domains such as tourism and education. Moreover, We can explore the possibility of using above techniques for various English Indian language translation. In future, we would also like to employ machine translation in detecting offensive languages from code-switched languages too (Mathur et al., 2018).

## References

What percentage of people in india speak english? https://tinyurl.com/indianlanguageStats. Accessed: 2018-08-27.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Andrew Donald Booth. 1955. Machine translation of languages, fourteen essays.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Siddhartha Ghosh, Sujata Thamke, et al. 2014. Translation of telugu-marathi and vice-versa using rule based machine translation. *arXiv preprint arXiv:1406.3969*.

Krupakar Hans and RS Milton. 2016. Improving the performance of neural machine translation involving morphologically rich languages. *arXiv preprint arXiv:1612.02482*.

Nadeem Jadoon Khan, Waqas Anwar, and Nadir Durrani. 2017. Machine translation approaches and survey for indian languages. *arXiv preprint arXiv:1701.04290*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. 2018. Detecting offensive tweets in hindi-english code-switched language. In *Proceedings of the International Workshop on Natural Language Processing for Social Media*, pages 18–26.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Raj Nath Patel, Prakash B Pimpale, et al. 2017. Mtil17: English to indian langauge statistical machine translation. *arXiv preprint arXiv:1708.07950*.

Amarnath Pathak and Partha Pakray. Neural machine translation for indian languages. *Journal of Intelligent Systems*.

Sree Harsha Ramesh and Krishna Prasad Sankaranarayanan. 2018. Neural machine translation for low resource languages using bilingual lexicon induced from comparable corpora. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 112–119.

Karthik Revanuru, Kaushik Turlapaty, and Shrisha Rao. 2017. Neural machine translation of indian languages. In *Proceedings of the 10th Annual ACM India Compute Conference on ZZZ*, pages 11–20. ACM.

Pramod Salunkhe, Aniket D Kadam, Shashank Joshi, Shuhas Patil, Devendrasingh Thakore, and Shrikant Jadhav. 2016. Hybrid machine translation for english to marathi: A research evaluation in machine translation:(hybrid translator). In *Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on*, pages 924–931. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Reshef Shilon. 2011. *Transfer-based Machine Translation between morphologically-rich and resource-poor languages: The case of Hebrew and Arabic*. Ph.D. thesis, Citeseer.

Harold Somers. 2003. An overview of ebmt. In *Recent advances in example-bas ed machine translation*, pages 3–57. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Fai Wong, Mingchui Dong, and Dongcheng Hu. 2006. Machine translation using constraint-based synchronous grammar. *Tsinghua Science and Technology*, 11(3):295–306.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

LoganathanRamasamy OndrejBojar ZdenekŽabokrtskỳ. 2012. Morphological processing for english-tamil statistical machine translation. In *24th International Conference on Computational Linguistics*, page 113.

# The Benefit of Pseudo-Reference Translations in Quality Estimation of MT Output

**Melania Duma and Wolfgang Menzel**
University of Hamburg
Natural Language Systems Division
{duma, menzel}@informatik.uni-hamburg.de

## Abstract

In this paper, a novel approach to Quality Estimation is introduced, which extends the method in (Duma and Menzel, 2017) by also considering pseudo-reference translations as data sources to the tree and sequence kernels used before. Two variants of the system were submitted to the sentence level WMT18 Quality Estimation Task for the English-German language pair. They have been ranked 4th and 6th out of 13 systems in the SMT track, while in the NMT track ranks 4 and 5 out of 11 submissions have been reached.

## 1 Introduction

The purpose of Quality Estimation (QE), as a subfield of Machine Translation (MT), is to allow the evaluation of MT output without the necessity of providing a reference translation. This would be extremely beneficial in the development cycle of a MT system, as it would permit fast and cost efficient evaluation phases. In the case of the previous Quality Estimation Shared Task (Bojar et al., 2017) together with the current campaign (Specia et al., 2018a), the purpose for the sentence level track was to predict the effort required in order to post-edit a candidate translation as measured by the Human-mediated Translation Edit Rate (HTER) (Snover et al., 2006) score.

In this paper an extension of the QE method introduced in (Duma and Menzel, 2017) is presented. Our earlier version of the metric was based on learning HTER scores using tree and sequence kernels. The kernel functions were applied not only on the source segments and the candidate translations, but also on the back-translations of the MT output into the source language. The back-translations were obtained using an online MT system.

The extension proposed in this paper uses the same input data. In addition, however, the kernel functions are defined to also consider pseudo-references as an additional source of evidence. The pseudo-references represent translations of the source segments into the target language and were obtained using the same online MT system as for the back-translation. By applying both the sequence and the tree kernels on the pseudo-references, we wanted to determine if an additional data source, even if artificially generated, would have a positive impact on our previous QE method. Throughout the rest of the paper we will refer to both the newly developed QE method as well as to its earlier version as Tree and Sequence Kernel Quality Estimation (*TSKQE*), but the variant under consideration will be marked through the use of subscripts together with superscripts.

This paper is organized as follows. In Section 2 related work is presented, focusing on kernel based QE methods. In the next section the implementation details for *TSKQE* are presented. This is followed by the evaluation setup and a discussion of the results. The paper concludes with future work ideas and final remarks.

## 2 Related work

The benefit of kernel functions has already been investigated in the context of Quality Estimation. In the work presented by (Hardmeier, 2011) and further expanded in (Hardmeier et al., 2012), tree kernel functions in addition to feature vectors are used to predict MT output quality. Both constituency and dependency parse trees were considered, with the Subset Tree Kernels (Collins and Duffy, 2001) being applied to the former and the Partial Tree Kernel (Moschitti, 2006a)(Moschitti, 2006b) to the latter. The evaluation results revealed that the integration of tree kernels can prove beneficial when compared to the strictly feature based QE systems.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 776–781
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64087

Tree kernels have also been applied in the work of (Kaljahi et al., 2014) and (Kaljahi, 2015), where a QE system is built based on Subset Tree Kernels applied for the constituency and dependency parse trees corresponding to the source and candidate translation. The kernels were also combined with a series of manually designed features, while SVM regression was used, in order to predict different automatic MT evaluation methods, like for example BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and METEOR (Denkowski and Lavie, 2014) scores.

The QE method introduced in (Duma and Menzel, 2017), *TSKQE*, is based on a linear combination between tree and sequence kernels. As a tree kernel the Partial Tree Kernel (PTK) is used, while for the sequence kernel, the Subsequence Kernel (SK) (Bunescu and Mooney, 2005) was chosen. Similarly to the previously mentioned QE methods, the kernels are applied to the source and candidate translations, but in addition also on a back-translation. The work presented in this paper builds on this method, by additionally using kernel functions for pseudo-references. Pseudo-references have been utilized before in the context of QE, but as a support for the generation of features, like for example in the work of (Soricut et al., 2012), (Shah et al., 2013) or (Scarton and Specia, 2014). In (Scarton and Specia, 2014) BLEU and TER were applied to the candidate translation and pseudo-references and their scores were used as additional features in the context of document level QE.

## 3 Method details

Different variants of *TSKQE* were defined in (Duma and Menzel, 2017) depending on the **level** where the kernel functions are applied (source segment, candidate translation or back-translation) and the **type** of kernel function (SK or PTK).

To indicate these distinctions we will use a notation system, where the level will be marked as a subscript attached to the *TSKQE* method name, with the possible values being *source* in case of the source segments, *basic* corresponding to both source segments and candidate translations, *back* for back-translations and *pseudo* corresponding to the newly introduced pseudo-references. In the case of the type, this will be marked as a superscript, with only two possible values, *sk* for the Sequence Kernel and *ptk* for the Partial Tree Kernel. For the variants where both kernel functions are used, the superscript will be left unfilled. Examples for this notation can be found in Tables 1 and 2.

*TSKQE* requires parsed input data, which was generated by means of the MATE parser (Bohnet, 2010), using English and German pre-trained models for tokenization, lemmatization, tagging and parsing itself [1]. The resulting dependency tree was further processed in order to remove the arc labels and encode all the syntactic information as tree nodes. For this, a variant of the Lexical-Centered-Tree (LCT) (Croce et al., 2011) method was applied, so that the dependency relation becomes the rightmost child of the dependency heads. For the generation of the pseudo-references and back-translations, the Google Translator Toolkit [2] was used.

The actual *TSKQE* models were built with the help of the Kernel-based Learning Platform (KeLP) library (Filice et al., 2015b) (Filice et al., 2015a), where various kernel functions and learning algorithms are integrated. For our experiments, we used the Support Vector Machine epsilon-Regression algorithm to learn the HTER scores, together with the PTK and SK implementations.

## 4 Evaluation

The evaluation was performed measuring the correlation between the *TSKQE* scores and the HTER gold standards. This was achieved by computing the Pearson correlation coefficient, which results in a number between -1 and 1. A score of 1 indicates that there is a perfect agreement between the two sets of scores, while a score of -1 would suggest a negative agreement. In addition to the Pearson coefficient, the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) were also calculated. For both these evaluation methods, the closer their score is to 0, the better the QE system should be considered.

The significance testing of the results was performed using the methodology presented in (Graham, 2015), which is based on pairwise testing using the Williams test (Williams, 1959). [3]

[1] All these models can be found at https://code.google.com/archive/p/matetools/downloads

[2] https://translate.google.com/toolkit

[3] The script used for computing the significance testing can be found at https://github.com/ygraham/mt-qe-eval.

| | SMT | | | NMT | | |
|---|---|---|---|---|---|---|
| **System** | **Pearson** | MAE | RMSE | **Pearson** | MAE | RMSE |
| $TSKQE^{sk}_{source}$ | 0.468 | 0.141 | 0.183 | 0.341 | 0.138 | 0.185 |
| $TSKQE^{sk}_{basic}$ | 0.517 | 0.136 | 0.176 | 0.387 | 0.136 | 0.181 |
| $TSKQE^{sk}_{basic+back}$ | 0.522 | 0.135 | 0.176 | 0.391 | 0.136 | 0.180 |
| $TSKQE^{sk}_{basic+pseudo}$ | 0.512 | 0.135 | 0.177 | 0.407 | 0.135 | 0.179 |
| $TSKQE^{sk}_{basic+back+pseudo}$ | 0.523 | 0.135 | 0.176 | 0.409 | 0.135 | 0.178 |
| $TSKQE^{ptk}_{source}$ | 0.440 | 0.142 | 0.186 | 0.361 | 0.133 | 0.181 |
| $TSKQE^{ptk}_{basic+back}$ | 0.517 | 0.136 | 0.176 | 0.376 | 0.136 | 0.181 |
| $TSKQE^{ptk}_{basic+pseudo}$ | 0.507 | 0.136 | 0.178 | 0.391 | 0.135 | 0.180 |
| $TSKQE^{ptk}_{basic+back+pseudo}$ | 0.517 | 0.135 | 0.176 | 0.392 | 0.135 | 0.180 |
| $TSKQE_{basic}$ | 0.532 | 0.134 | 0.175 | 0.395 | 0.135 | 0.180 |
| $TSKQE_{basic+back}$ | **0.537** | 0.133 | 0.174 | 0.400 | 0.136 | 0.180 |
| $TSKQE_{basic+pseudo}{}^{*}$ | 0.523 | 0.134 | 0.176 | 0.414 | 0.134 | 0.178 |
| $TSKQE_{basic+back+pseudo}{}^{*}$ | 0.534 | 0.133 | 0.174 | **0.417** | 0.135 | 0.178 |
| Baseline WMT | 0.359 | 0.147 | 0.195 | 0.264 | 0.129 | 0.184 |
| Baseline $TSKQE^{ptk}_{basic}$ | 0.509 | 0.135 | 0.177 | 0.371 | 0.135 | 0.181 |

Table 1: The results of the evaluation for the different TSKQE models.

In terms of the data sets, *TSKQE* was evaluated on the English-German datasets (Specia et al., 2018b) provided by the WMT18 Quality Estimation sentence level task. In contrast to the years before, the campaign offered two tracks for this language pair: in addition to the traditional one focused on SMT systems, another one considered the evaluation of an NMT system. Both tracks used translations from the IT domain, with the data consisting of tuples made up of the source segment, the candidate translation, the reference translation and the HTER score associated to that candidate translation. For the NMT system, 13,442 tuples were made available for the training, with an additional 1,000 tuples provided for development purposes. In the case of the SMT system, the training set was larger, consisting of 26,273 instances, with the same number of 1000 tuples made available for evaluation.

We compared the performance of *TSKQE* with a weak but also with a strong baseline. The former is represented by the QE system trained only on the 17 baseline features offered by the WMT18 QE campaign organizers. The features [4] have been regularly used over the past campaigns and include, for example, the number of tokens in the source sentence or the LM probability of the target sentence. We used these baseline features not only to build the baseline system, but also integrated them into *TSKQE* by means of a Radial Basis Function (RBF) kernel. For this purpose, we applied a Z-score standardization to rescale the feature values.

For the strong baseline, we considered a variant of one of the QE systems introduced by (Hardmeier et al., 2012), based on Partial Tree Kernels applied to the source segments and candidate translations. In our notation, this would correspond to the $TSKQE^{ptk}_{basic}$ notation.

The results of the evaluation for both the NMT and the SMT tracks are presented in Table 1. We highlighted in bold the highest Pearson values. Furthermore, we marked using an asterisk the two variants which we have chosen as our submissions

[4] A list of the baseline features can be found at https://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox_baseline_17

| NMT Models | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 0.002 |
| 2 | 0 | - | - | 0.072 | 0.119 | 0.22 | - | - | - | - | - | - | - | - | 0 |
| 3 | 0 | 0.257 | - | 0.051 | 0.079 | 0.133 | - | - | - | - | 0.478 | - | - | - | 0 |
| 4 | 0.082 | - | - | - | - | - | - | - | - | - | - | - | - | - | 0 |
| 5 | 0.031 | - | - | 0.215 | - | - | - | - | - | - | - | - | - | - | 0 |
| 6 | 0.019 | - | - | 0.141 | 0.229 | - | - | - | - | - | - | - | - | - | 0 |
| 7 | 0 | 0.06 | 0.32 | 0.021 | 0.015 | 0.059 | - | - | - | - | 0.357 | 0.394 | - | - | 0 |
| 8 | 0 | 0.063 | 0.054 | 0.013 | 0.01 | 0.014 | 0.231 | - | - | - | 0.238 | 0.241 | - | - | 0 |
| 9 | 0 | 0.003 | 0.041 | 0.006 | 0.007 | 0.02 | 0.066 | 0.227 | - | - | 0.104 | 0.13 | - | - | 0 |
| 10 | 0 | 0.004 | 0.002 | 0.005 | 0.005 | 0.01 | 0.053 | 0.095 | 0.331 | - | 0.082 | 0.088 | - | - | 0 |
| 11 | 0.002 | 0.408 | - | 0.022 | 0.01 | 0.066 | - | - | - | - | - | - | - | - | 0 |
| 12 | 0.002 | 0.388 | 0.497 | 0.024 | 0.021 | 0.015 | - | - | - | - | 0.437 | - | - | - | 0 |
| 13 | 0 | 0.002 | 0.014 | 0.001 | 0 | 0.002 | 0.005 | 0.058 | 0.085 | 0.25 | 0.009 | 0.021 | - | - | 0 |
| 14 | 0 | 0.002 | 0.001 | 0.001 | 0 | 0 | 0.006 | 0.004 | 0.1 | 0.086 | 0.009 | 0.006 | 0.326 | - | 0 |
| 15 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |

| SMT Models | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | - | - | - | 0.034 | - | - | - | - | - | - | - | - | - | - | 0 |
| 2 | 0 | - | - | 0 | 0.29 | 0.489 | - | - | 0.278 | - | 0.253 | 0.499 | - | - | 0 |
| 3 | 0 | 0.218 | - | 0 | 0.183 | 0.336 | - | - | 0.156 | - | 0.158 | 0.349 | - | - | 0 |
| 4 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 0 |
| 5 | 0.007 | - | - | 0 | - | - | - | - | - | - | 0.404 | - | - | - | 0 |
| 6 | 0.003 | - | - | 0 | 0.152 | - | - | - | 0.388 | - | 0.174 | - | - | - | 0 |
| 7 | 0 | 0.006 | 0.128 | 0 | 0.012 | 0.089 | - | - | 0.015 | 0.178 | 0.014 | 0.105 | 0.118 | - | 0 |
| 8 | 0 | 0.013 | 0.007 | 0 | 0.009 | 0.023 | 0.252 | - | 0.012 | 0.049 | 0.009 | 0.035 | 0.079 | 0.338 | 0 |
| 9 | 0.001 | - | - | 0 | 0.418 | - | - | - | - | - | 0.354 | - | - | - | 0 |
| 10 | 0 | 0.265 | 0.475 | 0 | 0.182 | 0.33 | - | - | 0.038 | - | 0.128 | 0.323 | - | - | 0 |
| 11 | 0.01 | - | - | 0 | - | - | - | - | - | - | - | - | - | - | 0 |
| 12 | 0.002 | - | - | 0 | 0.203 | 0.478 | - | - | 0.366 | - | 0.065 | - | - | - | 0 |
| 13 | 0 | 0.262 | 0.472 | 0 | 0.126 | 0.307 | - | - | 0.028 | 0.482 | 0.051 | 0.282 | - | - | 0 |
| 14 | 0 | 0.057 | 0.098 | 0 | 0.027 | 0.067 | 0.427 | - | 0.003 | 0.025 | 0.007 | 0.036 | 0.032 | - | 0 |
| 15 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |

$1 = TSKQE^{sk}_{source}$ $2 = TSKQE^{sk}_{basic}$ $3 = TSKQE^{sk}_{basic+back}$
$4 = TSKQE^{ptk}_{source}$ $5 = TSKQE^{ptk}_{basic}$ $6 = TSKQE^{ptk}_{basic+back}$
$7 = TSKQE_{basic}$ $8 = TSKQE_{basic+back}$ $9 = TSKQE^{sk}_{basic+pseudo}$
$10 = TSKQE^{sk}_{basic+back+pseudo}$ $11 = TSKQE^{ptk}_{basic+pseudo}$ $12 = TSKQE^{ptk}_{basic+back+pseudo}$
$13 = TSKQE_{basic+pseudo}$ $14 = TSKQE_{basic+back+pseudo}$ 15 = weak baseline

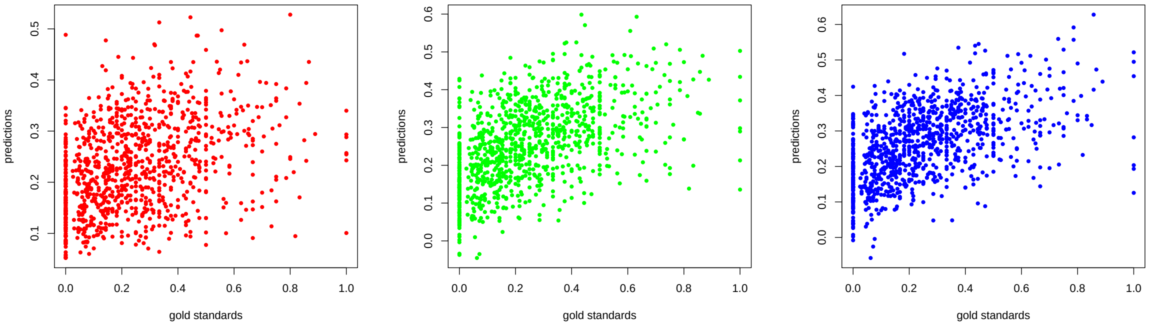
Table 2: Significance Williams test results.

to the WMT18 QE sentence level task. The results of the significance tests for two sets of *TSKQE* models are displayed in Table 2. Here, each table can be read as a matrix, where both the rows and columns correspond to the different *TSKQE* systems. The significance testing was performed only for the pairs of systems where the column model achieved a higher Pearson correlation than the row model. Otherwise, the cell was marked with a hyphen sign.

## 4.1 Discussion of the results

The results presented in Table 1 show that all the *TSKQE* variants outperform the weak baseline systems in terms of Pearson correlation. The same applies in the case of the strong baseline, with a few exceptions like the exclusively source based models. This result is not surprising, since the source based QE systems have access to no other input data except the source segments. The only information they receive about the candidate translation is the one contained in the baseline features.

Comparing the *TSKQE* variants based on pseudo-references with the other models, a noticeable improvement of the Pearson coefficients can be observed for the NMT system, while in the case of the SMT system the use of the pseudo-references brings no change or actually leads to a small drop in performance, which can be observed for example when comparing the *basic+pseudo* models to the *basic+back* ones. The significance tests reveal that the improvements, in the case of the NMT system, are statistically significant for the *basic+back+pseudo* models over the *basic+back* ones at a level of 0.05. In the case of the SMT system the differences between the *basic+back+pseudo* models and the *basic+back* ones are not statistically significant. In terms of



(a) SMT weak baseline model (b) SMT strong baseline model (c) SMT *TSKQE* model scores

Figure 1: Plots of the TSKQE and baseline model scores compared to the golden standards.

the best performing model, taking into account both MT systems, $TSKQE_{basic+back+pseudo}$, the SK and PTK based *TSKQE* variant which uses all the possible data sources, including the pseudo references, achieved on average the best correlation. These results suggest that the incorporation of the pseudo-references can be advantageous for building a high quality *TSKQE* system.

A further analysis of the results highlights the high quality of the SK based models. This is an important aspect to note, as it shows that even in the case of lower resourced language pairs, which might lack syntactic analysis tools, the SK based variants can still predict HTER scores with a comparable accuracy to the ones generated by the SK and PTK combination based models.

We also studied the degree of correlation between the predicted and the gold standard scores. Figure 1 shows the plots for the weak and the strong baseline models as well as for the $TSKQE_{basic+back+pseudo}$ model, all applied to the SMT data. [5]. Obviously, the weak baseline system encounters difficulties in predicting the HTER score as there is very little correlation between the two sets of scores. In case of the strong baseline, the predicted scores start to display a positive correlation with the gold ones, with this trend becoming even more evident in the case of the $TSKQE_{basic+back+pseudo}$ model.

## 5 Conclusions and future work

In this paper, we examined an extension of *TSKQE*, the sentence level QE method introduced in (Duma and Menzel, 2017). The evaluation results have not only confirmed the high quality of *TSKQE*, but they also showed that the use of pseudo-references as additional data sources for the kernel functions can be beneficial for the performance of *TSKQE*. Furthermore, the results indicate that *TSKQE* is robust against the choice of a particular MT paradigm producing comparably good results for both SMT and NMT systems.

In future work, we would like to extend the evaluation to include additional language pairs and domains. Another interesting line of research would be the use of constituency trees in addition to the dependency trees already explored to determine if these additional syntactic structures would be advantageous to the performance of *TSKQE*.

## References

Bernd Bohnet. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. *The 23rd International Conference on Computational Linguistics (COLING 2010)*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214.

Razvan Bunescu and Raymond Mooney. 2005. Subsequence Kernels for Relation Extraction. *Advances in Neural Information Processing Systems, Vol. 18: Proceedings of the 2005 Conference (NIPS)*.

Michael Collins and Nigel Duffy. 2001. Convolution Kernels for Natural Language. *Proceedings of NIPS 2001*, pages 625–632.

[5] The plots were obtained using the R language (R Core Team, 2014) and its packages

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured Lexical Similarity via Convolution Kernels on Dependency Trees. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1034–1046.

Michael Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Melania Duma and Wolfgang Menzel. 2017. UHH Submission to the WMT17 Quality Estimation Shared Task. In *Proceedings of the Second Conference on Machine Translation*, pages 556–561.

Simone Filice, Giuseppe Castellucci, Roberto Basili, Giovanni Da San Martino, and Alessandro Moschitti. 2015a. KeLP: a Kernel-based Learning Platform in Java. *The workshop on Machine Learning Open Source Software (MLOSS): Open Ecosystems*.

Simone Filice, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2015b. KeLP: a kernel-based learning platform for Natural Language Processing. *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 19–24.

Yvette Graham. 2015. Improving Evaluation of Machine Translation Quality Estimation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1804–1813.

Christian Hardmeier. 2011. Improving Machine Translation Quality Prediction with Syntactic Tree Kernels. *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 233–240.

Christian Hardmeier, Joakim Nivre, and Jorg Tiedemann. 2012. Tree Kernels for Machine Translation Quality Estimation. *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 109–113.

Rasoul Kaljahi. 2015. The Role of Syntax and Semantics in Machine Translation and Quality Estimation of Machine-translated User-generated Content. *PhD Thesis*.

Rasoul Kaljahi, Jennifer Foster, Raphael Rubino, and Johann Roturier. 2014. Quality Estimation of English-French Machine Translation: A Detailed Study of the Role of Syntax. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2052–2063.

Alessandro Moschitti. 2006a. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. *Proceedings of the 17th European Conference on Machine Learning*.

Alessandro Moschitti. 2006b. Making Tree Kernels Practical for Natural Language Learning. *Proceedings of the Eleventh International Conference of the European Association for Computational Linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

R Core Team. 2014. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Carolina Scarton and Lucia Specia. 2014. Document-level translation quality estimation: exploring discourse and pseudo-references. In *The 17th Annual Conference of the European Association for Machine Translation*, pages 101–108.

Kashif Shah, Trevor Cohn, and Lucia Specia. 2013. An Investigation on the Effectiveness of Features for Translation Quality Estimation. In *Proceedings of the Machine Translation Summit*, volume 14, pages 167–174. Citeseer.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Radu Soricut, Nguyen Bach, and Ziyuan Wang. 2012. The SDL Language Weaver Systems in the WMT12 Quality Estimation Shared Task. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 145–151. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón F. Astudillo, and André Martins. 2018a. Findings of the WMT 2018 Shared Task on Quality Estimation. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Lucia Specia, Varvara Logacheva, Frederic Blain, Ramon Fernandez, and André Martins. 2018b. WMT18 Quality Estimation Shared Task Training and Development Data. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (UFAL), Faculty of Mathematics and Physics, Charles University.

Evan James Williams. 1959. *Regression analysis*, volume 14. Wiley New York.

# Supervised and Unsupervised Minimalist Quality Estimators: Vicomtech's Participation in the WMT 2018 Quality Estimation Task

**Thierry Etchegoyhen** and **Eva Martínez Garcia** and **Andoni Azpeitia**
Vicomtech
Mikeletegi Pasalekua, 57
Donostia / San Sebastián, Gipuzkoa, Spain
{tetchegoyhen, emartinez, aazpeitia}@vicomtech.org

## Abstract

We describe Vicomtech's participation in the WMT 2018 shared task on quality estimation, for which we submitted minimalist quality estimators. The core of our approach is based on two simple features: lexical translation overlaps and language model cross-entropy scores. These features are exploited in two system variants: uMQE is an unsupervised system, where the final quality score is obtained by averaging individual feature scores; sMQE is a supervised variant, where the final score is estimated by a Support Vector Regressor trained on the available annotated datasets. The main goal of our minimalist approach to quality estimation is to provide reliable estimators that require minimal deployment effort, few resources, and, in the case of uMQE, do not depend on costly data annotation or post-editing. Our approach was applied to all language pairs in sentence quality estimation, obtaining competitive results across the board.

## 1 Introduction

Quality Estimation (QE) refers to the task of estimating the quality of machine translation output without access to reference translations (Blatz et al., 2004), which are not always available for a given domain or language pair, and are costly to produce.

Typical approaches are based on supervised machine learning models using a large array of features, as exemplified by the standard QUEST baseline (Specia et al., 2013), whose base version employs 17 features that include n-gram language model perplexity scores, lexical translation probabilities, number of source tokens and average number of translations per source word, among others. In recent years, QE models based on neural network approaches have significantly improved the state of the art, as shown for instance by the results obtained in the WMT 2016 and WMT 2017 shared tasks (Kim and Lee, 2016; Kim et al., 2017; Martins et al., 2017).

Despite recent progress, the vast number of potential domains and language pairs is a challenging aspect for a practical use of quality estimation systems. First, most approaches to QE rely on annotated data, typically based on human post-editing, which are costly to produce. Additionally, the best performing approaches based on neural networks (e.g., Kim et al., 2017) require large volumes of parallel training corpora, a resource which is only available for a small number of language pairs nowadays.

To tackle these challenges, we designed a minimalist approach to quality estimation, to which we will refer as MQE, based on two features: a lexical translation overlap measure to model translation accuracy[1] and a measure based on cross-entropy scores according to a target language model. No external tools or large computational resources are needed in this approach, which can be used in the two variants described below.

uMQE is an unsupervised variant, where the final quality score is obtained by averaging individual feature scores. The system was designed to provide reliable estimators in the numerous use cases where no training data are available to train supervised QE models. To our knowledge, little attention has been paid to this type of approaches, with two main published approaches: Moreau and Vogel (2012) estimate the quality of machine-translated output against external sets of n-grams and evaluate several variants of n-gram similarity, whereas Popovic (2012) proposes an unsupervised method based on the arithmetic combination of scores provided by language models and IBM1 models, trained on morphemes as well as part-of-speech tags. On the WMT 2012 datasets, neither

[1] Also referred to as *adequacy*.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 782–787
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64088

approach performed better than the QUEST baseline. In this paper, we show that our own unsupervised approach can outperform the supervised baselines, without the use of additional resources such as part-of-speech taggers or morphological analysers.

sMQE is a supervised variant, where the final score is estimated by a Support Vector Regressor trained on the available machine translation output annotated with HTER scores. The goal of this approach is to enable a fast deployment of supervised quality estimators that outperform other supervised approaches with more complex setups, such as the QUEST baselines with 17 features, while using minimal resources. Contrary to uMQE, for which only rank correlation is meaningful, the supervised variant can be evaluated on both ranking and scoring tasks.

The paper is organised as follows: Section 2 describes the core MQE approach and the computation of the supervised and unsupervised variants; Section 3 describes the experimental setup for the WMT 2018 shared task on sentence quality estimation; Section 4 presents our results on the test sets in all four language pairs and domains; finally, Section 5 draws conclusions from this work.

## 2 MQE

Minimally, quality estimation involves determining the accuracy (or adequacy) of a translation, i.e. how much of the source information is represented in the translation, and its fluency, i.e. the correctness of the generated sentence as a target language sequence. MQE directly models these two aspects, to the exclusion of any other property of the source and target sentence pairs. We describe our measures of accuracy and fluency in turn in the next sections.

### 2.1 Accuracy

To measure accuracy, we adapted the approach in (Etchegoyhen and Azpeitia, 2016), which has proved highly successful in identifying parallel sentences in large sets of comparable corpora (Azpeitia et al., 2017, 2018). Their method is based on Jaccard similarity (Jaccard, 1901) over lexical sets, with additional set expansion operations to address named entities and morphological variation. We describe their core methodology below and our adaptations for the quality estimation task.

Let $s_i$ and $s_j$ be two tokenised and truecased sentences in languages $l_1$ and $l_2$, respectively, $S_i$ and $S_j$ the multisets[2] of tokens in $s_i$ and $s_j$, respectively, $T_{ij}$ the multiset of lexical translations into $l_2$ for all tokens in $S_i$, and $T_{ji}$ the multiset of lexical translations into $l_1$ for all tokens in $S_j$.

Lexical translations are computed from sentences $s_i$ and $s_j$ by retaining the $k$-best translations for each word, as determined by the ranking obtained from the translation probabilities given by symmetrised IBM2 word alignment models (Brown et al., 1993).[3] The multisets $T_{ij}$ and $T_{ji}$ that comprise these $k$-best lexical translations are then expanded by means of the following operations:[4]

1. For each element $x$ in the set difference $T'_{ij} = T_{ij} - S_j$ (respectively $T'_{ji} = T_{ji} - S_i$), and each element $y$ in $S_j$ (respectively $S_i$), if $x$ and $y$ share a common prefix of more than $n$ characters, the prefix is added to both $T_{ij}$ and $S_j$ (respectively $T_{ji}$ and $S_i$). Longest common prefix matching is meant to capture morphological variation via minimal computation.

2. Numbers and capitalised truecased tokens not found in the lexical translation tables are added to the expanded translation multisets $T_{ij}$ and $T_{ji}$. This operation addresses named entities, which are likely to be missing from translation tables trained on different domains.

3. The NULL token is added to the source and target token multisets, in order to address words that have covert translations, as indicated by the presence of the NULL element among their $k$-best translation options.

With source and target sets as defined above, we compute translation accuracy between sentence $s_i$ and translation $s_j$ as in Equation 1:

[2] We employ multisets instead of sets as in the original approach, to account for multiple token occurrences, as the quality estimation task is more likely to be sensitive to missing occurrences than the alignment task. Multiset intersection and union are based on positive minimums and maximums, respectively.

[3] The actual probabilities are not used beyond determining the ranking, as in the original approach. We depart from their implementation by using IBM2 models instead of IBM4, a change motivated by the similar results we obtained with both types of models and the faster training of the former.

[4] The first two are based on the original approach, while the third was added by us for the experiments reported here.

$$acc(s_i, s_j) = \frac{1}{2}\left(\frac{|T_{ij} \cap S_j|}{|S_j|} + \frac{|T_{ji} \cap S_i|}{|S_i|}\right) \quad (1)$$

Accuracy is thus defined as the mean of the overlap similarity coefficients obtained between sentence token sets and expanded lexical translation sets in both directions.[5] Apart from the use of multisets and the introduction of the NULL element, the main change to the original metric is using overlap instead of Jaccard similarity, as the former provided better results in preliminary experiments.

Although originally meant to identify parallel sentences in comparable corpora, this simple metric applies naturally to any task involving lexical translations and provides an efficient method to model accuracy.

### 2.2 Fluency

The standard approach to measuring the fluency of word sequences in a given language is by means of language models. Although n-gram modelling has been the dominant approach in the last two decades, continuous space language models have become a new standard and have been notably used for the quality estimation task, providing improvements in supervised feature-based frameworks (Shah et al., 2015b). For the experiments presented here, we nonetheless used n-gram language modelling as a first approach, as they provided the best results overall in preliminary experiments and require comparatively fewer computational resources to be trained.

As a measure of fluency, we take the inverse of the per word cross-entropy for each machine-translated sentence. The fluency score is thus computed according to Equation 2, where $P(w_i)$ is short for $P(w_i|w_{i-(k-1)}, \ldots, w_{i-1})$, i.e. the conditional probability of the $i$-th word given its $k$ preceding words in sentence $s_j$ of length $n$.

$$flc(s_j) = \frac{1}{-\frac{1}{n}\sum_{i=1}^{n} logP(w_i)} \quad (2)$$

Thus, the higher the cross-entropy, the lower the fluency score. Although simple, measures computed via n-gram language models, such as cross-entropy or the monotonically-related perplexity, have been shown to be reliable indicators of translation quality estimation (Shah et al., 2015a).

### 2.3 MQE Variants

For the unsupervised uMQE variant, we assume that task-related annotated data are not available to optimise feature weighting,[6] and thus simply take the arithmetic mean of the two scores as our final quality estimation score. Since the two scores are not in similar ranges, we perform min/max feature rescaling on all scores prior to combining the features. The final quality estimation score for a source $s_i$ and translation $s_j$ is computed as in Equation 3, with rescaled features $acc^r$ and $flc^r$.

$$uMQE(s_i, s_j) = \frac{acc^r(s_i, s_j) + flc^r(s_j)}{2} \quad (3)$$

For the supervised variant, sMQE, we used the annotated datasets provided for the WMT 2018 QE task and trained a Support Vector Regressor (SVR) with a Radial Basis Function kernel on the two features, using the default parameters provided by the scikit-learn toolkit[7] (C=1.0, $\epsilon$=0.1, and $\gamma$=0.5 for 2 features):

$$sMQE(s_i, s_j) = SVR([acc(s_i, s_j), flc(s_j)]) \quad (4)$$

## 3 Experimental Setup

We submitted results from our two system variants in all language pairs for sentence-level QE, using the same models for both variants in each case. To train the IBM2 and language models, we selected corpora available for the WMT shared tasks for each specific domain and language pair. For English-German, in the IT domain, we used the training data from the WMT 2016 IT translation task, the WMT 2017 QE task and the WMT 2018 PE task; given the low amounts of data in each individual corpus, we also merged the data from the technical manuals of *OpenOffice* and *KDE4* available in the OPUS repository (Tiedemann, 2012). For German-English, in the biomedical domain, we used the UFAL medical corpus[8], combined with the training data from the WMT 2018 QE

[5] Note that the denominator in a set-based overlap measure is the smallest of the two sets being compared, which in our case is always the token set.

[6] Such datasets were available for the WMT 2018 shared task, but we opted to ignore them in order to test the uMQE variant under its intended unsupervised conditions of use.

[7] http://scikit-learn.org/

[8] https://ufal.mff.cuni.cz/ufal_medical_corpus

| LANG | DOMAIN | MT | SYSTEM | SPEARMAN | RK$^{\rho}$ | PEARSON | RK$^{r}$ | MAE | RMSE |
|---|---|---|---|---|---|---|---|---|---|
| EN-DE | IT | SMT | UMQE | 0.3787 | 12/15 | – | – | – | – |
| EN-DE | IT | SMT | UMQE* | 0.4042 | 7/15 | – | – | – | – |
| EN-DE | IT | SMT | SMQE | 0.3993 | 7/15 | 0.3969 | 9/14 | 0.1855 | 0.2248 |
| EN-DE | IT | NMT | UMQE | 0.3999 | 10/13 | – | – | – | – |
| EN-DE | IT | NMT | UMQE* | 0.4542 | 6/13 | – | – | – | – |
| EN-DE | IT | NMT | SMQE | 0.4439 | 6/13 | 0.3716 | 9/12 | 0.2063 | 0.2421 |
| DE-EN | BIOMED | SMT | UMQE | 0.5694 | 5/11 | – | – | – | – |
| DE-EN | BIOMED | SMT | SMQE | 0.6003 | 4/11 | 0.6521 | 4/10 | 0.1182 | 0.1547 |
| EN-LV | BIOMED | SMT | UMQE | 0.3979 | 3/8 | – | – | – | – |
| EN-LV | BIOMED | SMT | SMQE | 0.4061 | 2/8 | 0.4612 | 2/7 | 0.1318 | 0.1767 |
| EN-LV | BIOMED | NMT | UMQE | 0.5403 | 3/7 | – | – | – | – |
| EN-LV | BIOMED | NMT | SMQE | 0.5686 | 2/7 | 0.5787 | 2/6 | 0.1461 | 0.1938 |
| EN-CS | IT | SMT | UMQE | 0.4196 | 6/9 | – | – | – | – |
| EN-CS | IT | SMT | SMQE | 0.4219 | 5/9 | 0.3904 | 7/8 | 0.1638 | 0.2122 |

Table 1: Results on the WMT 2018 test sets

task. For English-Latvian, also in the biomedical domain, we used the available EMEA corpus along with the training data from the WMT 2018 QE task and the additional data provided for this language pair in this year's QE task. Finally, for English-Czech in the IT domain, we used the *train-techdoc* section of the CzEng17 dataset available for the WMT 2018 translation task, along with the QE training data and the additional data provided for the WMT 2018 QE task.

Sentences were tokenised and truecased with the scripts available in the Moses toolkit (Koehn et al., 2007), with truecasing models trained on the data described above. For English-Czech, we experimented with BPE segmentation (Sennrich et al., 2016) to overcome data sparseness issues, training BPE models with a maximum of 30.000 merge operations and segmenting all corpora accordingly for this language pair.

All IBM2 models were trained with the FASTALIGN toolkit (Dyer et al., 2013), and all language models are of order 5 trained with the KENLM toolkit (Heafield, 2011) on the target language data. For the accuracy metric, minimal prefix length was set to 4 and $k$-best translation lists limited to 4 candidates.

## 4 Results

The results on the WMT 2018 test sets are shown in Table 1.[9] Overall the results were satisfactory for both variants of such a simple minimalist approach. For English-Latvian for instance, sMQE and uMQE ranked in second and third place, respectively; for German-English, the two variants ranked fourth and fifth, respectively. Our worst results were obtained for English-Czech and English-German, although for the latter our system still ranked in the top half among competing systems on the ranking task, and, except for the scoring task in EN-CS, both variants outperformed the baselines across the board. The relatively worse results obtained for these two language pairs can be tied to data sparseness issues affecting our simple fluency feature based on n-gram cross-entropy.

The results obtained by uMQE were overall slightly lower than those obtained by the supervised sMQE variant, although the small number of features available to train the SVR for the latter was not expected to lead to major improvements. Our unsupervised approach gave satisfactory results, performing significantly better in most cases than the supervised baseline with 17 features. We view this as an important result, considering the vast number of domains and language pairs where no training data are available to opt for a supervised approach.

Even in cases where task-related data exist, the amount of available parallel corpora in a given language pair might not be sufficient to train sophisticated neural quality estimators. In such cases, the sMQE variant can also provide a reliable alternative to perform quality estimation under minimal resources.

The approach is also fairly simple to implement and deploy, and does not require external tagging or parsing tools which may not be available for

[9] In the table, RK$^{\rho}$ and RK$^{r}$ indicate the ranking of the system among all participants in terms of Spearman and Pearson correlation, respectively. Note that the official uMQE results for English-German are based on erroneous submissions and we submitted the correct version after the deadline via CODALAB to obtain the expected scores. The correct version, using the same models as for sMQE, is denoted by uMQE* and we refer to the results of this submission in the discussion relative to this language pair.

many languages. It is thus a highly portable alternative which may be the simplest and most efficient option in a significant number of scenarios, with results that outperform the standard supervised baseline across the board.

## 5 Conclusions

We have described our participation in the WMT 2018 shared task on quality estimation, which included both supervised and unsupervised variants of a minimalist approach to the task. Both variants are based on two simple measures of accuracy, computed from lexical translation overlap, and fluency, computed from inverse cross-entropy scores of an n-gram language model.

Our main goal was to evaluate systems that can be efficiently deployed for the large number of language pairs and domains where there are either no annotated data at all to train a supervised system, or insufficient amounts of parallel corpora to adequately train the currently best performing neural quality estimators. Additionally, our approach requires no external tools such as part-of-speech taggers or syntactic parsers, unlike other competing approaches, and is thus both simpler to deploy and readily available for languages where such tools are not available at all.

We view the obtained results as satisfactory, with both variants outperforming the supervised baselines overall and being placed among the five best systems in two of the four language pairs. In future work, we will evaluate the use of continuous space language models to address data sparseness issues in the two language pairs where more complex morphology limits the contribution of an n-gram-based fluency feature. We will also explore variants of the accuracy measure and evaluate in more details the aspects that can be better modelled under the proposed minimalist approach to quality estimation.

**Acknowledgements** This work was supported by the Department of Economic Development and Competitiveness of the Basque Government via project QUALES (KK-2017/00094). We wish to thank the anonymous WMT 2018 reviewers for their detailed and helpful reviews of this work. Opinions, errors and omissions are our own.

## References

Andoni Azpeitia, Thierry Etchegoyhen, and Eva Martínez Garcia. 2017. Weighted Set-Theoretic Alignment of Comparable Sentences. In *Proceedings of the Tenth Workshop on Building and Using Comparable Corpora*, pages 41–45. Association for Computational Linguistics.

Andoni Azpeitia, Thierry Etchegoyhen, and Eva Martínez Garcia. 2018. Extracting Parallel Sentences from Comparable Corpora with STACC Variants. In *Proceedings of the Eleventh Workshop on Building and Using Comparable Corpora*, pages 48–52. European Language Resources Association (ELRA).

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 315–321. Association for Computational Linguistics.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational linguistics*, 19(2):263–311.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Thierry Etchegoyhen and Andoni Azpeitia. 2016. Set-Theoretic Alignment for Comparable Corpora. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1: Long Papers, pages 2009–2018.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Paul Jaccard. 1901. Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:241 – 272.

Hyun Kim and Jong-Hyeok Lee. 2016. Recurrent Neural Network based Translation Quality Estimation. In *Proceedings of the First Conference on Machine Translation*, pages 787–792. Association for Computational Linguistics.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared*

*Task Papers*, pages 562–568. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL*, pages 177–180. Association for Computational Linguistics.

André F. T. Martins, Fabio Kepler, and Jose Monteiro. 2017. Unbabel's participation in the wmt17 translation quality estimation shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 569–574. Association for Computational Linguistics.

Erwan Moreau and Carl Vogel. 2012. Quality estimation: an experimental study using unsupervised similarity measures. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 120–126. Association for Computational Linguistics.

Maja Popovic. 2012. Morpheme- and pos-based IBM1 and language model scores for translation quality estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 133–137.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Kashif Shah, Trevor Cohn, and Lucia Specia. 2015a. A bayesian non-linear method for feature selection in machine translation quality estimation. *Machine Translation*, 29(2):101–125.

Kashif Shah, Raymond WM Ng, Fethi Bougares, and Lucia Specia. 2015b. Investigating continuous space language models for machine translation quality estimation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1073–1078.

Lucia Specia, Kashif Shah, Jose GC de Souza, Trevor Cohn, and Fondazione Bruno Kessler. 2013. QuEst–a translation quality estimation framework. In *Proceedings of the 51st ACL: System Demonstrations*, pages 79–84.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th Language Resources and Evaluation Conference*, pages 2214–2218.

# Contextual Encoding for Translation Quality Estimation

**Junjie Hu, Wei-Cheng Chang, Yuexin Wu, Graham Neubig**
Language Technologies Institute, Carnegie Mellon University
{junjieh, wchang2, yuexinw, gneubig}@cs.cmu.edu

## Abstract

The task of word-level quality estimation (QE) consists of taking a source sentence and machine-generated translation, and predicting which words in the output are correct and which are wrong. In this paper, propose a method to effectively encode the local and global contextual information for each target word using a three-part neural network approach. The first part uses an embedding layer to represent words and their part-of-speech tags in both languages. The second part leverages a one-dimensional convolution layer to integrate local context information for each target word. The third part applies a stack of feed-forward and recurrent neural networks to further encode the global context in the sentence before making the predictions. This model was submitted as the CMU entry to the WMT2018 shared task on QE, and achieves strong results, ranking first in three of the six tracks.[1]

## 1 Introduction

Quality estimation (QE) refers to the task of measuring the quality of machine translation (MT) system outputs without reference to the gold translations (Blatz et al., 2004; Specia et al., 2013). QE research has grown increasingly popular due to the improved quality of MT systems, and potential for reductions in post-editing time and the corresponding savings in labor costs (Specia, 2011; Turchi et al., 2014). QE can be performed on multiple granularities, including at word level, sentence level, or document level. In this paper, we focus on quality estimation at word level, which is framed as the task of performing binary classification of translated tokens, assigning "OK" or "BAD" labels.

Early work on this problem mainly focused on hand-crafted features with simple regression/classification models (Ueffing and Ney, 2007; Biçici, 2013). Recent papers have demonstrated that utilizing recurrent neural networks (RNN) can result in large gains in QE performance (Martins et al., 2017). However, these approaches encode the context of the target word by merely concatenating its left and right context words, giving them limited ability to control the interaction between the local context and the target word.

In this paper, we propose a neural architecture, Context Encoding Quality Estimation (CEQE), for better encoding of context in word-level QE. Specifically, we leverage the power of both (1) convolution modules that automatically learn local patterns of surrounding words, and (2) hand-crafted features that allow the model to make more robust predictions in the face of a paucity of labeled data. Moreover, we further utilize stacked recurrent neural networks to capture the long-term dependencies and global context information from the whole sentence.

We tested our model on the official benchmark of the WMT18 word-level QE task. On this task, it achieved highly competitive results, with the best performance over other competitors on English-Czech, English-Latvian (NMT) and English-Latvian (SMT) word-level QE task, and ranking second place on English-German (NMT) and German-English word-level QE task.

## 2 Model

The QE module receives as input a tuple $\langle s, t, \mathcal{A} \rangle$, where $s = s_1, \ldots, s_M$ is the source sentence, $t = t_1, \ldots, t_N$ is the translated sentence, and $\mathcal{A} \subseteq \{(m, n) | 1 \leq m \leq M, 1 \leq n \leq N\}$ is a set of word alignments. It predicts as output a sequence $\hat{y} = y_1, \ldots, y_N$, with each $y_i \in \{\text{BAD}, \text{OK}\}$. The

[1] Our software is available at https://github.com/junjiehu/CEQE.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 788–793
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64089

overall architecture is shown in Figure 1

CEQE consists of three major components: (1) embedding layers for words and part-of-speech (POS) tags in both languages, (2) convolution encoding of the local context for each target word, and (3) encoding the global context by the recurrent neural network.

### 2.1 Embedding Layer

Inspired by (Martins et al., 2017), the first embedding layer is a vector representing each target word $t_j$ obtained by concatenating the embedding of that word with those of the aligned words $s_{\mathcal{A}(:,t_j)}$ in the source. If a target word is aligned to multiple source words, we average the embedding of all the source words, and concatenate the target word embedding with its average source embedding. The immediate left and right contexts for source and target words are also concatenated, enriching the local context information of the embedding of target word $t_j$. Thus, the embedding of target word $t_j$, denoted as $\mathbf{x}_j$, is a $6d$ dimensional vector, where $d$ is the dimension of the word embeddings. The source and target words use the same embedding parameters, and thus identical words in both languages, such as digits and proper nouns, have the same embedding vectors. This allows the model to easily identify identical words in both languages. Similarly, the POS tags in both languages share the same embedding parameters. Table 1 shows the statistics of the set of POS tags over all language pairs.

| Language Pairs | Source | Target |
|---|---|---|
| En-De (SMT) | 50 | 57 |
| En-De (NMT) | 49 | 58 |
| De-En | 58 | 50 |
| En-Lv (SMT) | 140 | 38 |
| En-Lv (NMT) | 167 | 43 |
| En-Cz | 440 | 57 |

Table 1: Statistics of POS tags over all language pairs

### 2.2 One-dimensional Convolution Layer

The main difference between the our work and the neural model of Martins et al. (2017) is the one-dimensional convolution layer. Convolutions provide a powerful way to extract local context features, analogous to implicitly learning $n$-gram features. We now describe this integral part of our model.

After embedding each word in the target sentence $\{t_1, \dots, t_j, \dots, t_N\}$, we obtain a matrix of embeddings for the target sequence,

$$\mathbf{x}_{1:N} = \mathbf{x}_1 \oplus \mathbf{x}_2 \dots \oplus \mathbf{x}_N,$$

where $\oplus$ is the column-wise concatenation operator. We then apply one-dimensional convolution (Kim, 2014; Liu et al., 2017) on $\mathbf{x}_{1:N}$ along the target sequence to extract the local context of each target word. Specifically, a one-dimensional convolution involves a filter $\mathbf{w} \in \mathbb{R}^{hk}$, which is applied to a window of $h$ words in target sequence to produce new features.

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b),$$

where $b \in \mathbb{R}$ is a bias term and $f$ is some functions. This filter is applied to each possible window of words in the embedding of target sentence $\{\mathbf{x}_{1:h}, \mathbf{x}_{2:h+1}, \dots, \mathbf{x}_{N-h+1:N}\}$ to produce features

$$\mathbf{c} = [c_1, c_2, \dots, c_{N-h+1}].$$

By the padding proportionally to the filter size $h$ at the beginning and the end of target sentence, we can obtain new features $\mathbf{c}_{pad} \in \mathbb{R}^N$ of target sequence with output size equals to input sentence length $N$. To capture various granularities of local context, we consider filters with multiple window sizes $\mathcal{H} = \{1, 3, 5, 7\}$, and multiple filters $n_f = 64$ are learned for each window size.

The output of the one-dimensional convolution layer, $C \in \mathbb{R}^{N \times |\mathcal{H}| \cdot n_f}$, is then concatenated with the embedding of POS tags of the target words, as well as its aligned source words, to provide a more direct signal to the following recurrent layers.

### 2.3 RNN-based Encoding

After we obtain the representation of the source-target word pair by the convolution layer, we follow a similar architecture as (Martins et al., 2017) to refine the representation of the word pairs using feed-forward and recurrent networks.

1. Two feed-forward layers of size 400 with rectified linear units (ReLU; Nair and Hinton (2010));

2. One bi-directional gated recurrent unit (BiGRU; Cho et al. (2014)) layer with hidden size 200, where the forward and backward hidden states are concatenated and further normalized by layer normalization (Ba et al., 2016).

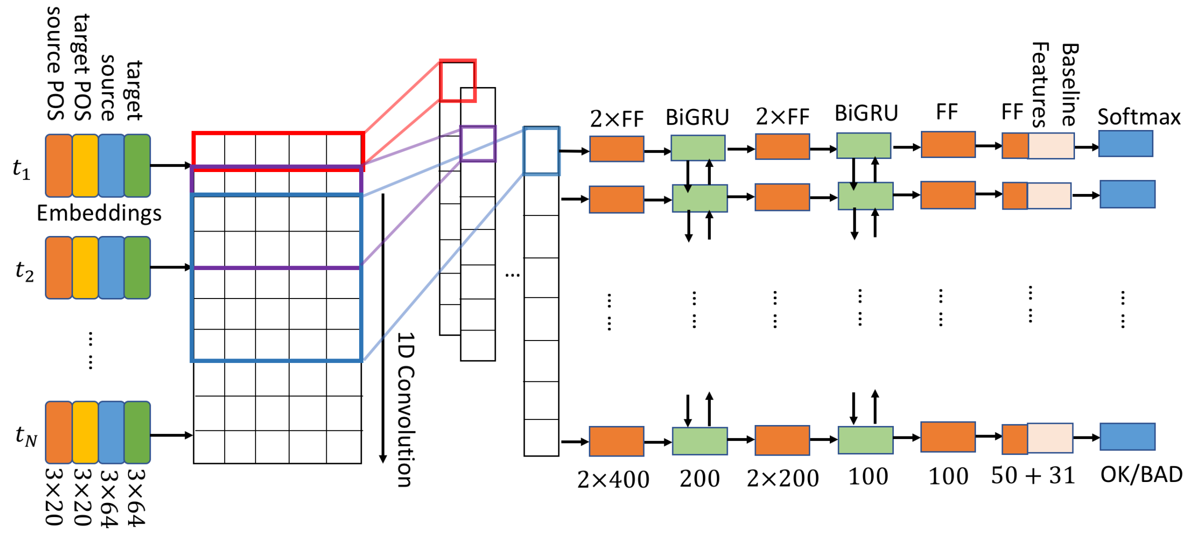


Figure 1: The architecture of our model, with the convolutional encoder on the left, and stacked RNN on the right.

| Category | Description |
|---|---|
| Binary | target word is a stopword |
| Binary | target word is a punctuation mark |
| Binary | target word is a proper noun |
| Binary | target word is a digit |
| Float | backoff behavior of ngram $w_{i-2}$ $w_{i-1}$ $w_i$ ($w_i$ is the target word) |
| Float | backoff behavior of ngram $w_{i-1}$ $w_i$ $w_{i+1}$ |
| Float | backoff behavior of ngram $w_i$ $w_{i+1}$ $w_{i+2}$ |
| One-hot | highest order of ngram that includes target word and its left context |
| One-hot | highest order of ngram that includes target word and its right context |
| One-hot | highest order of ngram that includes source word and its left context |
| One-hot | highest order of ngram that includes source word and its right context |

Table 2: Baseline Features

3. Two feed-forward layers of hidden size 200 with rectified linear units;

4. One BiGRU layer with hidden size 100 using the same configuration of the previous BiGRU layer;

5. Two feed-forward layers of size 100 and 50 respectively with ReLU activation.

We concatenate the 31 baseline features extracted by the Marmot[2] toolkit with the last 50 feed-forward hidden features. The baseline features are listed in Table 2. We then apply a softmax layer on the combined features to predict the binary labels.

## 3 Training

We minimize the binary cross-entropy loss between the predicted outputs and the targets. We train our neural model with mini-batch size 8 using Adam (Kingma and Ba, 2015) with learning rate 0.001 and decay the learning rate by multiplying 0.75 if the F1-Multi score on the validation set decreases during the validation. Gradient norms are clipped within 5 to prevent gradient explosion for feed-forward networks or recurrent neural networks. Since the training corpus is rather small, we use dropout (Srivastava et al., 2014) with probability 0.3 to prevent overfitting.

## 4 Experiment

We evaluate our CEQE model on the WMT2018 Quality Estimation Shared Task[3] for word-level English-German, German-English, English-Czech, and English-Latvian QE. Words in all languages are lowercased. The evaluation metric is the multiplication of F1-scores for the "OK" and "BAD" classes against the true labels. F1-score is the harmonic mean of precision and recall. In Table 3, our model achieves the best performance on three out of six test sets in the WMT 2018 word-level QE shared task.

### 4.1 Ablation Analysis

In Table 4, we show the ablation study of the features used in our model on English-German, German-English, and English-Czech. For each

[2] https://github.com/qe-team/marmot

[3] http://statmt.org/wmt18/quality-estimation-task.html

| Language Pairs | F1-BAD | F1-OK | F1-Multi | Rank |
|---|---|---|---|---|
| En-De (SMT) | 0.5075 | 0.8394 | 0.4260 | 3 |
| En-De (NMT) | 0.3565 | 0.8827 | 0.3147 | 2 |
| De-En | 0.4906 | 0.8640 | 0.4239 | 2 |
| En-Lv (SMT) | 0.4211 | 0.8592 | 0.3618 | 1 |
| En-Lv (NMT) | 0.5192 | 0.8268 | 0.4293 | 1 |
| En-Cz | 0.5882 | 0.8061 | 0.4741 | 1 |

Table 3: Best performance of our model on six datasets in the WMT2018 word-level QE shared task on the leader board (updated on July 27th 2018)

language pair, we show the performance of CEQE without adding the corresponding components specified in the second column respectively. The last row shows the performance of the complete CEQE with all the components. As the baseline features released in the WMT2018 QE Shared Task for English-Latvian are incomplete, we train our CEQE model without using such features. We can glean several observations from this data:

1. Because the number of “OK” tags is much larger than the number of “BAD” tags, the model is easily biased towards predicting the “OK” tag for each target word. The F1-OK scores are higher than the F1-BAD scores across all the language pairs.

2. For German-English, English Czech, and English-German (SMT), adding the baseline features can significantly improve the F1-BAD scores.

3. For English-Czech, English-German (SMT), and English-German (NMT), removing POS tags makes the model more biased towards predicting “OK” tags, which leads to higher F1-OK scores and lower F1-BAD scores.

4. Adding the convolution layer helps to boost the performance of F1-Multi, especially on English-Czech and English-Germen (SMT) tasks. Comparing the F1-OK scores of the model with and without the convolution layer, we find that adding the convolution layer help to boost the F1-OK scores when translating from English to other languages, i.e., English-Czech, English-German (SMT and NMT). We conjecture that the convolution layer can capture the local information more effectively from the aligned source words in English.

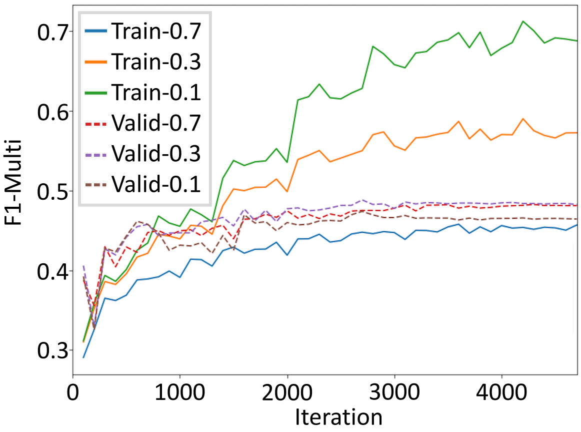


Figure 2: Effect of the dropout rate during training.

## 5 Case Study

Table 5 shows two examples of quality prediction on the validation data of WMT2018 QE task for English-Czech. In the first example, the model without POS tags and baseline features is biased towards predicting “OK” tags, while the model with full features can detect the reordering error. In the second example, the target word “panelu” is a variant of the reference word “panel”. The target word “znaky” is the plural noun of the reference “znak”. Thus, their POS tags have some subtle differences. Note the target word “zmnit” and its aligned source word “change” are both verbs. We can observe that POS tags can help the model capture such syntactic variants.

### 5.1 Sensitivity Analysis

During training, we find that the model can easily overfit the training data, which yields poor performance on the test and validation sets. To make the model more stable on the unseen data, we apply dropout to the word embeddings, POS embeddings, vectors after the convolutional layers and the stacked recurrent layers. In Figure 2, we examine the accuracies dropout rates in $[0.1, 0.3, 0.7]$. We find that adding dropout alleviates overfitting issues on the training set. If we reduce the dropout rate to $0.1$, which means randomly setting some values to zero with probability $0.1$, the training F1-Multi increases rapidly and the validation F1-multi score is the lowest among all the settings. Preliminary results proved best for a dropout rate of $0.3$, so we use this in all the experiments.

| Language Pairs | Method | F1-BAD | F1-OK | F1-Multi |
|---|---|---|---|---|
| De-En | - (Convolution + POS + features) | 0.4774 | 0.8680 | 0.4144 |
| | - (POS + features) | 0.4948 | 0.8474 | 0.4193 |
| | - features | 0.5095 | **0.8735** | 0.4450 |
| | - POS | 0.4906 | 0.8640 | 0.4239 |
| | CEQE | **0.5233** | 0.8721 | **0.4564** |
| En-Cz | - (Convolution + POS + features) | 0.5748 | 0.7622 | 0.4381 |
| | - (POS + features) | 0.5628 | 0.8000 | 0.4502 |
| | - features | 0.5777 | 0.7997 | 0.4620 |
| | - POS | 0.5192 | **0.8268** | 0.4293 |
| | CEQE | **0.5884** | 0.7991 | **0.4702** |
| En-De (SMT) | - (Convolution + POS + features) | 0.4677 | 0.8038 | 0.3759 |
| | - (POS + features) | 0.4768 | 0.8166 | 0.3894 |
| | - features | 0.4902 | 0.8230 | 0.4034 |
| | - POS | 0.5047 | **0.8431** | 0.4255 |
| | CEQE | **0.5075** | 0.8394 | **0.4260** |
| En-De (NMT) | - (Convolution + POS + features) | 0.3545 | 0.8396 | 0.2976 |
| | - (POS + features) | 0.3404 | 0.8752 | 0.2979 |
| | - features | **0.3565** | 0.8827 | **0.3147** |
| | - POS | 0.3476 | **0.8948** | 0.3111 |
| | CEQE | 0.3481 | 0.8835 | 0.3075 |

Table 4: Ablation study on the WMT18 Test Set

## 6 Conclusion

In this paper, we propose a deep neural architecture for word-level QE. Our framework leverages a one-dimensional convolution on the concatenated word embeddings of target and its aligned source words to extract salient local feature maps. In additions, bidirectional RNNs are applied to capture temporal dependencies for better sequence prediction. We conduct thorough experiments on four language pairs in the WMT2018 shared task. The proposed framework achieves highly competitive results, outperforms all other participants on English-Czech and English-Latvian word-level, and is second place on English-German, and German-English language pairs.

## Acknowledgements

The authors thank Andre Martins for his advice regarding the word-level QE task.

This work is sponsored by Defense Advanced Research Projects Agency Information Innovation Office (I2O). Program: Low Resource Languages for Emergent Incidents (LORELEI). Issued by DARPA/I2O under Contract No. HR0011-15-C0114. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

## References

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Ergun Biçici. 2013. Referential translation machines for quality estimation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 343–351.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 315. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. 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)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

| | |
|---|---|
| Source | specify the scope of blending options : |
| MT | určete rozsah prolnutí voleb : |
| Reference | určete rozsah voleb prolnutí : |
| no POS & features | určete rozsah prolnutí voleb : |
| CEQE | určete rozsah prolnutí voleb : |
| Source | use the Character panel and Paragraphs panel to change the appearance of text . |
| MT | pomocí panelu znaky a odstavce , chcete - li změnit vzhled textu . |
| Reference | použijte panel znak a panel odstavce , chcete - li změnit vzhled textu . |
| no POS & features | pomocí panelu znaky a odstavce , chcete - li zmnit vzhled textu . |
| CEQE | pomocí panelu znaky a odstavce , chcete - li změnit vzhled textu . |

Table 5: Examples on WMT2018 validation data. The source and translated sentences, the reference sentences, the predictions of the CEQE without and with POS tags and baseline features are shown. Words predicted as OK are shown in green, those predicted as BAD are shown in red, the difference between the translated and reference sentences are shown in blue.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124. ACM.

André Martins, Marcin Junczys-Dowmunt, Fabio Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics*, 5:205–218.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Lucia Specia. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 73–80.

Lucia Specia, Kashif Shah, Jose GC Souza, and Trevor Cohn. 2013. Quest-a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Marco Turchi, Antonios Anastasopoulos, José GC de Souza, and Matteo Negri. 2014. Adaptive quality estimation for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 710–720.

Nicola Ueffing and Hermann Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40.

# Sheffield Submissions for the WMT18 Quality Estimation Shared Task

**Julia Ive**[1], **Carolina Scarton**[2], **Frédéric Blain**[2] and **Lucia Specia**[2]
[1]IoPPN, Kings College London, UK
[2]DCS, University of Sheffield, UK
julia.ive@kcl.ac.uk
{c.scarton, f.blain, l.specia}@sheffield.ac.uk

## Abstract

In this paper we present the University of Sheffield submissions for the WMT18 Quality Estimation shared task. We discuss our submissions to all four sub-tasks, where ours is the only team to participate in all language pairs and variations (37 combinations). Our systems show competitive results and outperform the baseline in nearly all cases.

## 1 Introduction

Quality Estimation (QE) predicts the quality of Machine Translation (MT) when automatic evaluation or human assessment is not possible (typically at system run-time). QE is mainly addressed as a supervised Machine Learning problem with QE models trained using labelled data. These labels differ for different tasks, for example, binary labels for fine-grained predictions (e.g. OK/BAD for words or phrases) and continuous measurements of quality for coarse-grained levels (e.g. HTER (Snover et al., 2006) for sentences).

For this year's shared task, post-edited (PE) and manually annotated data were provided. They cover four levels of predictions: sentence-level (task 1), word-level (task 2), phrase-level (task 3) and document-level (task 4), over five language pairs: English into German, Latvian, Czech and French, as well as German-English. For the first time, these data contain translations produced by neural MT (NMT) systems. Such translations are known to be more fluent but less adequate (Toral and Sánchez-Cartagena, 2017).

For tasks 2 and 3, this year's edition introduces a new task variant of predicting missing words in the translations. Thus two additional prediction types are required: (i) binary labels for gaps in the translation to indicate whether one or more tokens are missing from a certain position, and (ii) binary labels for words in source sentences to indicate which of these words lead to incorrect words in the translations.

We participated with two different systems, both available in the DeepQuest[1] toolkit (Ive et al., 2018):

- **SHEF-PT:** an in-house re-implementation of the POSTECH system (Kim et al., 2017b), and
- **SHEF-bRNN:** a bidirectional recurrent neural network (bRNN) system.

We participated in all sub-tasks and submitted a total of 74 predictions (37 per system).

## 2 Systems Description

Our light-weight neural QE approach is based on simple encoders and requires no pre-training (bRNN). We compare its performance to the performance of our re-implementation of the state-of-the-art neural QE approach of Kim et al. (2017a,b) (POSTECH), which uses a complex architecture and requires resource-intensive pre-training.

### 2.1 Architecture

Following current best practices in neural sequence-to-sequence modelling (Sutskever et al., 2014; Bahdanau et al., 2015), our bRNN approach employs encoders using recurrent neural networks (RNNs). Encoders encode input into an internal representation used to make classification decisions. bRNN representations at a given level rely on representations from more fine-grained levels (i.e. sentences for document, and words for phrase and sentence).

bRNN uses two bi-directional RNNs to learn the representation of the <source, MT> sentence pair. Source and MT RNNs are trained independently.

[1] https://sheffieldnlp.github.io/deepQuest

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 794–800
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64090

The two representations are then combined via concatenation. For word-level QE, those representations (sequences of hidden states $h_j$ associated with words) can be used directly to make classification decisions. A sentence vector is a weighted sum of word vectors as generated by an attention mechanism. Another output layer takes this sentence vector as input and produces real-value sentence-level quality scores.

For phrase-level QE, we have modified the architecture described above. It takes a three-dimensional MT input (batch length × sentence length in phrases × phrase length in words).[2] Concatenation of source and MT sentence representations, as performed in our word- and sentence-level architecture, will require source inputs to be three-dimensional as well. However, as the phrase alignments are not provided with the task, three-dimensional source inputs can not be formed without an additional approximation.[3] Instead, we follow best practices of NMT (Bahdanau et al., 2015) and implement its standard encoder-decoder architecture. The encoder creates source representations using a bidirectional RNN, at each timestep the decoder produces a word representation taking into account not only the previously produced representations, but also the sum of source word representations weighted by an attention mechanism.[4] This process can be interpreted as defining word alignments: the resulting decoder representations contain information on both MT words and respective parts of the source attended at each timestep. Each phrase representation can be computed out of word vectors: average, maximum, sum, etc. The resulting representations are provided to the output layer, as illustrated in Figure 1.

Our document-level framework is a wrapper over sentence QE approaches. It uses a bidirectional RNN to summarize sentence-level representations as document-level representations used for regression.

More details on the architecture and implementation of our sentence and document-level models can be found in Ive et al. (2018).

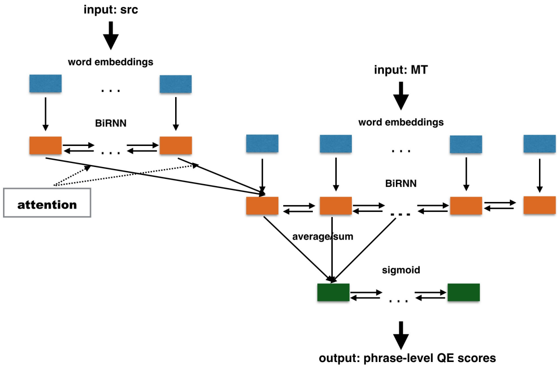


Figure 1: bRNN phrase-level QE architecture.

## 2.2 Implementation Details

To train POSTECH's predictor, we used the corresponding parts of the in-domain corpora provided by the organisers for the corresponding languages (≈ 2M sentences were selected randomly per language pair). The only exception was EN-LV for which we had less than 2M sentences in the corpus. Therefore, we combined the in-domain corpus with the Europarl (version 8)[5] and EMEA corpus.[6] This totaled in 1,241,615 EN-LV sentences.

For the word and phrase-level tasks, we tackled prediction of MT error tags, source tags and MT gaps separately. For predicting source tags, we built models by swapping source and MT inputs. POSTECH's predictors were then trained with swapped source and target inputs. For predicting gaps, we added a dummy word at the beginning of each MT sentence to match the count of gap tags per line.

We experimented with phrase-level representations and created them by computing the sum or the average of composing word vectors. To optimise the usage of computational resources, in each experiment we fixed the size of a phrase in words to the upper quartile of the respective distribution in the training data.

For the document-level QE, we experimented with sentence-level representations coming from both bRNN and POSTECH architectures.

For our POSTECH-based document-level models, we experimented with predictors trained on a

[2] Note that other architectural choices may lead to, for instance, two-dimensional inputs (batch length × phrase length in words). A representation of each MT phrase may be created without taking the rest of the translated sentence into account.

[3] For instance, we may assume that translation of a phrase relies on the whole source sentence. Thus, a three-dimensional input can be formed by simply repeating each source sentence along the second axis to match respective counts of phrases in each MT sentence.

[4] Note that Jhaveri et al. (2018) also use this architecture for sentence-level QE.

[5] http://www.statmt.org/wmt17/translation-task.html

[6] http://opus.nlpl.eu/EMEA.php

| | SHEF-PT | | | SHEF-bRNN | | | Baseline | | |
|---|---|---|---|---|---|---|---|---|---|
| | $r$ | MAE | $\rho$ | $r$ | MAE | $\rho$ | $r$ | MAE | $\rho$ |
| EN-DE – SMT | **0.487** | **0.132** | **0.510** | 0.366 | 0.139 | 0.378 | 0.365 | 0.140 | 0.381 |
| EN-DE – NMT | 0.377 | 0.131 | 0.468 | **0.381** | 0.130 | **0.480** | 0.287 | **0.129** | 0.420 |
| EN-LV – SMT | 0.375 | 0.141 | 0.329 | **0.396** | **0.138** | 0.332 | 0.353 | 0.155 | **0.348** |
| EN-LV – NMT | **0.463** | 0.166 | 0.446 | 0.421 | 0.172 | 0.409 | 0.444 | **0.163** | **0.458** |
| EN-CS | **0.533** | **0.150** | **0.537** | 0.501 | 0.157 | 0.506 | 0.394 | 0.165 | 0.414 |
| DE-EN | **0.554** | **0.130** | **0.501** | 0.482 | 0.143 | 0.443 | 0.332 | 0.151 | 0.325 |

Table 1: Evaluation of our systems for task 1 on the test set. We show scores of Pearson's $r$ correlation, MAE and Spearman's $\rho$ correlation.

part of the English–French Europarl (version 7),[7] as well as on an in-domain corpus (described in Section 3.4). As mentioned before, our document-level QE system is a modular architecture wrapping over any sentence-level QE model. We took advantage of this modularity and also attempted multi-task learning (MTL). We pre-trained the weights of sentence-level modules (both `bRNN` and `POSTECH`) to predict Multidimensional Quality Metrics (MQM)[8] scores for sentences (more details in Section 3.4).

## 3 Tasks Participation

The four QE tasks correspond to different levels of quality prediction: sentence-level (task 1), word-level (task 2 and 3a), phrase-level (task 3b) and document-level (task 4). For each prediction level, different language pairs and system outputs are provided. Below we provide a detailed description of the datasets together with the results for our submitted systems for each of these tasks.

### 3.1 Task 1: Sentence-level QE

Four language pairs are available for sentence-level scoring and ranking:

- EN-DE: sentences on the IT domain, with MT from either an SMT ($26,273$ training / $1,000$ development / $1,000$ test) or an NMT ($13,442$ training / $1,000$ development / $1,000$ test) system,
- EN-LV: sentences on the life sciences domain, with MT from either an SMT ($11,251$ training / $1,000$ development / $1,000$ test) or an NMT ($12,936$ training / $1,000$ development / $1,000$ test) system,
- EN-CS: sentences on the IT domain, with MT from an SMT system ($40,254$ training / $1,000$ development / $1,000$ test), and
- DE-EN: sentences on the life sciences domain, with MT from an SMT system ($25,963$ training / $1,000$ development / $1,000$ test).

In summary, there are six data setting variants and the quality score for prediction is HTER in all of them. For each variant in this task we submitted two systems: SHEF-PT and SHEF-bRNN. For the ranking evaluation, we rank sentences using the predicted HTER outputted by our systems.

Following the shared task setup, Pearson's $r$ correlation coefficient is used as the primary evaluation metric for the scoring task (with Mean Absolute Error – MAE – as the secondary metric), whilst Spearman's $\rho$ rank correlation coefficient is used as metric for the ranking task. The task baseline systems are Support Vector Machine (SVM) models trained with 17 baseline features from QuEst++ (Specia et al., 2015).

We show the official results in Table 1. Both our systems outperform the baseline for all the language pairs according to the main evaluation metric ($r$). SHEF-bRNN is better than SHEF-PT only for EN-DE – NMT and EN-LV – SMT. These may be cases where `bRNN` is able to better capture the fluency of high-quality MT by encoding it directly as sequences rather than assessing it word for word as `POSTECH`. On the official development set,[9] EN-DE – NMT and EN-LV – SMT translations have the best overall quality (on average HTER=0.17 versus HTER=0.28 for the rest of the systems).

### 3.2 Task 2: Word-level QE

Task 2 uses the same datasets as task 1. Target words are assigned a binary label (OK or BAD) based on the alignments between MT and post-edits extracted by the TER tool. In this year's edition, the organisers have also proposed the predic-

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

[8] http://www.qt21.eu/mqm-definition/definition-2015-12-30.html

[9] The organisers have not provided the gold labels for the test set.

| TRG words prediction | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | SHEF-PT | | | SHEF-bRNN | | | Baseline | | |
| | F1 BAD | F1 OK | F1-MULT | F1 BAD | F1 OK | F1-MULT | F1 BAD | F1 OK | F1-MULT |
| EN-DE – SMT | **0.508** | 0.846 | **0.430** | 0.453 | 0.811 | 0.367 | 0.412 | **0.882** | 0.363 |
| EN-DE – NMT | 0.335 | 0.869 | 0.291 | **0.351** | 0.863 | **0.303** | 0.197 | **0.918** | 0.181 |
| EN-LV – SMT | **0.416** | 0.869 | **0.361** | 0.409 | 0.860 | 0.351 | 0.381 | **0.905** | 0.345 |
| EN-LV – NMT | **0.519** | 0.809 | 0.420 | 0.503 | 0.828 | 0.416 | 0.487 | **0.864** | **0.421** |
| EN-CS | **0.556** | 0.796 | 0.443 | 0.554 | 0.792 | 0.439 | 0.534 | **0.834** | **0.445** |
| DE-EN | **0.485** | 0.874 | 0.424 | 0.446 | 0.871 | 0.389 | **0.485** | **0.902** | **0.437** |
| SRC words prediction | | | | | | | | | |
| | SHEF-PT | | | SHEF-bRNN | | | Baseline | | |
| | F1 BAD | F1 OK | F1-MULT | F1 BAD | F1 OK | F1-MULT | F1 BAD | F1 OK | F1-MULT |
| EN-DE – SMT | **0.422** | 0.799 | 0.337 | 0.414 | **0.821** | **0.340** | - | - | - |
| EN-DE – NMT | 0.314 | 0.841 | 0.264 | **0.330** | **0.865** | **0.286** | - | - | - |
| EN-LV – SMT | 0.351 | **0.859** | 0.302 | **0.357** | 0.857 | **0.306** | - | - | - |
| EN-LV – NMT | **0.444** | **0.814** | **0.361** | **0.444** | 0.800 | 0.355 | - | - | - |
| EN-CS | **0.493** | 0.799 | 0.394 | 0.490 | **0.811** | **0.398** | - | - | - |
| DE-EN | **0.392** | **0.887** | **0.348** | 0.366 | 0.875 | 0.320 | | | |
| Gaps prediction | | | | | | | | | |
| | SHEF-PT | | | SHEF-bRNN | | | Baseline | | |
| | F1 BAD | F1 OK | F1-MULT | F1 BAD | F1 OK | F1-MULT | F1 BAD | F1 OK | F1-MULT |
| EN-DE – SMT | **0.294** | **0.962** | **0.282** | 0.271 | 0.955 | 0.259 | - | - | - |
| EN-DE – NMT | 0.110 | 0.984 | 0.108 | **0.121** | **0.985** | **0.119** | - | - | - |
| EN-LV – SMT | **0.141** | 0.968 | **0.136** | 0.118 | **0.975** | 0.115 | - | - | - |
| EN-LV – NMT | **0.130** | **0.965** | **0.126** | 0.119 | 0.944 | 0.113 | - | - | - |
| EN-CS | 0.171 | **0.977** | 0.167 | **0.179** | 0.972 | **0.174** | - | - | - |
| DE-EN | **0.210** | **0.970** | **0.204** | 0.200 | 0.966 | 0.193 | - | - | - |

Table 2: Evaluation of our systems for task 2 on the test set. We show scores of F1-MULT, F1 for the OK class and F1 for the BAD class.

tion of gaps and source words quality. According to the TER alignment, all source words aligned to a target word will receive the same tag as the target word. For annotating gaps, a gap tag is placed after each token and in the beginning of the sentence. A gap tag will be BAD if one or more words were expected to appear in the gap, and OK otherwise.

Task 2 has 18 variants, for each of them we again submitted two systems: SHEF-PT and SHEF-bRNN.

The primary evaluation metric of task 2 is F1-MULT: multiplication of F1-scores for the OK and BAD classes. F1-scores of OK and BAD classes are used as secondary metrics. The baseline system for the target word predictions is a Conditional Random Fields (CRF) model trained with word-level baseline features from the Marmot (Logacheva et al., 2016) toolkit. There are no baseline systems for the prediction of gaps or source word issues.

Table 2 shows the official results. For prediction of target words, SHEF-PT is the best for EN-DE – SMT, EN-LV – SMT and EN-LV – NMT. SHEF-bRNN is the best for EN-DE – NMT. This confirms our previous conclusion that `bRNN` better captures the fluency of high-quality MT (cf. Section 3.1). For source words and gaps prediction, SHEF-bRNN and SHEF-PT show similar performance across language pairs.

To get a closer insight into the performance of our models, we manually analysed results for the official EN-DE – SMT/NMT development sets. For those two systems either SHEF-PT, or SHEF-bRNN performs the best respectively. Our observations suggest that, because of pre-training, SHEF-PT better captures SMT adequacy (cf. examples in Table 3; the term “screen readers” is correctly translated by the SMT system into German as “Bildschirmlesehilfen” and correctly marked as OK by SHEF-PT, but incorrectly marked as BAD by SHEF-bRNN). SHEF-bRNN better captures NMT fluency: e.g. only the word “Transparenzeffekte” correctly marked as BAD from the first part of the NMT translation in Table 3 vs. the context of this word marked as BAD by SHEF-PT.

### 3.3 Task 3: Phrase-level QE

This task considers a subset of the English-German SMT data from task 1 (Section 3.1). Here, the MT output has been manually anno-

| | |
|---|---|
| SRC | to make your content accessible to screen readers , avoid using these modes . |
| PE | um den Inhalt für Bildschirmlesehilfen zugänglich zu machen , vermeiden Sie diese Modi . |
| **SMT** | um den Inhalt für Bildschirmlesehilfen zugänglich machen , vermeiden Sie diese Modi . |
| gold | OK OK OK OK OK OK OK OK OK OK OK OK OK |
| PT | OK OK OK OK OK OK OK OK OK OK OK OK OK |
| bRNN | OK OK OK OK **BAD BAD BAD BAD BAD BAD BAD BAD** OK |
| SRC | besides applying transparency effects to single objects , you can apply them to groups . |
| PE | Sie können Transparenzeffekte nicht nur auf einzelne Objekte , sondern auch auf Gruppen anwenden . |
| **NMT** | Sie können nicht nur Transparenzeffekte auf einzelne Objekte anwenden , sondern auch auf Gruppen anwenden . |
| gold | OK OK OK OK BAD OK OK OK BAD OK OK OK OK OK OK OK |
| PT | **BAD BAD BAD BAD** BAD **BAD** OK OK **OK** OK **BAD BAD BAD BAD** OK **BAD** |
| bRNN | OK OK OK OK BAD OK OK OK **OK** OK OK **BAD** OK OK OK OK |

Table 3: Examples of prediction errors for task 2 on the EN-DE – SMT/NMT development sets

| | SHEF-PT | | | SHEF-bRNN | | | Baseline | | |
|---|---|---|---|---|---|---|---|---|---|
| | F1 BAD | F1 OK | F1-MULT | F1 BAD | F1 OK | F1-MULT | F1 BAD | F1 OK | F1-MULT |
| TRG words | **0.3338** | 0.8250 | **0.2754** | 0.3253 | 0.8235 | 0.2679 | 0.2714 | **0.9099** | 0.2469 |
| Gaps | **0.2730** | 0.8775 | **0.2396** | 0.2631 | **0.8785** | 0.2312 | - | - | - |
| SRC words | **0.5048** | **0.8137** | **0.4108** | 0.4920 | 0.7916 | 0.3895 | - | - | - |

Table 4: Evaluation of our systems for task 3a on the test set. We show scores of F1-MULT, F1 for the OK class and F1 for the BAD class.

| SHEF-PT | | | | |
|---|---|---|---|---|
| | F1 BAD | F1 OK | F1-MULT | F1 BAD_w_o |
| TRG phrases | 0.2294 | 0.8059 | 0.1849 | 0.0794 |
| Gaps | **0.1073** | 0.9349 | **0.1003** | - |
| SHEF-ATT-SUM | | | | |
| | F1 BAD | F1 OK | F1-MULT | F1 BAD_w_o |
| TRG phrases | 0.2881 | 0.7614 | 0.2194 | **0.1146** |
| Gaps | 0.1028 | **0.9416** | 0.0968 | - |
| Baseline | | | | |
| | F1 BAD | F1 OK | F1-MULT | F1 BAD_w_o |
| TRG phrases | **0.3919** | **0.9152** | **0.3584** | 0.0194 |
| Gaps | - | - | - | - |

Table 5: Evaluation of our systems for task 3b on the test set. We show scores of F1-MULT, F1 for the OK class, F1 for the BAD class and F1 for the BAD_word_order class.

tated at the phrase level with four labels: OK, BAD, BAD_word_order and BAD_omission, with the phrase boundaries defined by the SMT decoder. The last two labels are new to this task. They indicate whether a phrase is in an incorrect position in the sentence, or one or more word(s) are missing in a certain position, respectively. The subtasks of predicting gaps and source phrases quality were proposed similarly to task 2 (cf. Section 3.2).

The subtask data are provided with word-level segmentation. Task 3 is therefore divided into two subtasks 3a and 3b, for word- and phrase-level predictions, respectively.

**Task3a – word-level prediction** Word-level labels have been produced as follows: each word has been labelled according to the phrase it belongs to (i.e. as either OK, BAD or BAD_word_order); gaps have been labelled as either OK or BAD_omission. The evaluation metrics for this subtask are similar to task 2.

The official results are reported in Table 4. Our two systems outperform the baseline for the target words prediction, while there are no other results for gaps and source words predictions.

**Task3b – phrase-level prediction** In addition to the usual binary labels (OK and BAD), this subtask considers the BAD_word_order label. To tackle the phrase-level challenge, we implemented a new model as part of deepQuest (cf. Section 2). The submitted SHEF-ATT-SUM system takes the sum of composing word vectors to create phrase vectors used for regression. This configuration performed the best on the official development set.

The official results are reported in Table 5. While we perform better than the baseline for task 3a, we are not able to beat it at the phrase level. We believe this is because the dataset is too small to train a competitive neural model. There are no other results for gaps prediction.[10]

[10] We did not participate to the source phrases prediction task, since the phrase alignments were not provided by the organisers.

### 3.4 Task 4: Document-level QE

Task 4 consists in predicting document-level quality scores for MT of product reviews from the Amazon Product Reviews dataset (He and McAuley, 2016). For this task, a selection of Sports and Outdoors product titles and descriptions were machine translated from English into French. The MT system used is a state-of-the-art NMT system. The machine translated documents were annotated with word-level MQM information. The MQM taxonomy has three coarse-grained classes: accuracy, fluency and style. Each error was classified into one of the fine-grained classes within a main class and also according to its severity: minor (it does not change the meaning of the source), major (the meaning was changed by the incorrect word) or critical (besides changing the meaning the error results in a negative effect, e.g. the translation can be seen as offensive).

Document-level scores were devised as follows using the information about the errors and their severities:

$$score = 100 * (1.0 - T_{severity} * \frac{1.0}{N}) \qquad (1)$$

where $T_{severity}$ is the sum of the severity weights of all errors in a given document (predefined as minor $= 1.0$, major $= 5.0$ and critical $= 10$) and $N$ is the total number of words in this document.

For training, development and testing, $1,000$, 200 and 269 documents were made available, respectively. The baseline is an SVM model trained with 15 baseline document-level features from QuEst++. Evaluation is done in terms of Pearson's $r$ correlation scores.

Since the MQM scores are at the word level, Equation 1 can also be used to extract scores for sentences. We exploit this feature and create MTL systems trained to predict both sentence and document-level scores. We submitted two systems officially and also report three additional systems. Our systems are listed below, where systems with an * are the official submissions:

- *SHEF-PT (in-domain): POSTECH system pre-trained with in-domain data extracted from the English–French part[11] of the Gigaword corpus,[12]
- SHEF-PT (out-domain): POSTECH system pre-trained with the Europarl data,
- SHEF-bRNN: our bRNN system for document-level QE,
- SHEF-MTL-PT (in-domain): multi-task POSTECH pre-trained with the in-domain data, and
- *SHEF-MTL-bRNN: multi-task bRNN.

Table 6 shows the evaluation of our systems on the test set in terms of Pearson's $r$ and MAE. The baseline is considerably strong, achieving over 0.5 of correlation and the lowest MAE (56.09). SHEF-PT (in-domain) and SHEF-MTL-PT (in-domain) are the only systems that outperform the baseline. Note that the SHEF-MTL-bRNN system achieved results close to the baseline, even though it does not use any external resources (unlike the SHEF-PT systems and the baseline).

| | $r$ | MAE |
|---|---|---|
| SHEF-PT (in-domain) | **0.534** | 56.23 |
| SHEF-PT (out-domain) | 0.511 | 57.55 |
| SHEF-bRNN | 0.468 | 57.58 |
| SHEF-MTL-PT (in-domain) | 0.521 | 56.60 |
| SHEF-MTL-bRNN | 0.473 | 56.59 |
| Baseline | 0.512 | **56.09** |

Table 6: Evaluation of our systems for task 4 on the test set. We show scores of Pearson's $r$ correlation and MAE.

## 4 Conclusions

We presented our systems submitted to the WMT18 QE shared task. We experimented with two different architectures: our re-implementation of the POSTECH system (SHEF-PT) and our bRNN (bi-directional RNNs) approach (SHEF-bRNN). Although SHEF-PT is better than SHEF-bRNN for the majority of the task variants, SHEF-bRNN is still a competitive system and, given its simplicity and independence from external resources, it can be seen as a good alternative for low-resource languages. In addition, it is worth mentioning that SHEF-bRNN requires considerably less training time than SHEF-PT, which may better fit certain scenarios.

## Acknowledgments

Carolina Scarton is supported by the EC project SIMPATICO (H2020-EURO-6-2015, grant number 692819). Frédéric Blain is supported by the Amazon Academic Research Awards program.

[11] https://catalog.ldc.upenn.edu/LDC2011T10

[12] ≈300K segments were extracted, using XenC (Rousseau, 2013), as having the best perplexity according to a language model trained on a selection of the English in-domain Amazon reviews (≈200K segments).

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 507–517.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018. DeepQuest: a framework for neural-based quality estimation. In *Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics: Technical Papers*. The COLING 2017 Organizing Committee.

Nisarg Jhaveri, Manish Gupta, and Vasudeva Varman. 2018. Translation quality estimation for indian languages. In *Proceedings of th 21st International Conference of the European Association for Machine Translation (EAMT)*.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017a. Predictor-Estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 17(1):3:1–3:22.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017b. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation (WMT)*, pages 562–568.

Varvara Logacheva, Chris Hokamp, and Lucia Specia. 2016. MARMOT: A toolkit for translation quality estimation at the word level. In *Tenth International Conference on Language Resources and Evaluation (LREC)*, pages 3671–3674.

Anthony Rousseau. 2013. XenC: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, (100):73–82.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level Translation Quality Prediction with QuEst++. In *The 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*, Beijing, China.

Ilya Sutskever, Oriol Vinyals, and Quoc V. V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112.

Antonio Toral and Víctor M Sánchez-Cartagena. 2017. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. *arXiv preprint arXiv:1701.02901*.

# UAlacant machine translation quality estimation at WMT 2018: a simple approach using phrase tables and feed-forward neural networks

**Miquel Esplà-Gomis  Felipe Sánchez-Martínez  Mikel L. Forcada**
Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03071 Alacant, Spain
{mespla,fsanchez,mlf}@dlsi.ua.es

## Abstract

We describe the Universitat d'Alacant submissions to the word- and sentence-level machine translation (MT) quality estimation (QE) shared task at WMT 2018. Our approach to word-level MT QE builds on previous work to mark the words in the machine-translated sentence as *OK* or *BAD*, and is extended to determine if a word or sequence of words need to be inserted in the gap after each word. Our sentence-level submission simply uses the edit operations predicted by the word-level approach to approximate TER. The method presented ranked first in the sub-task of identifying insertions in gaps for three out of the six datasets, and second in the rest of them.

## 1 Introduction

This paper describes the Universitat d'Alacant submissions to the word- and sentence-level machine translation (MT) quality estimation (QE) shared task at WMT 2018 (Specia et al., 2018). Our approach is an extension of a previous approach (Esplà-Gomis et al., 2015a,b; Esplà-Gomis et al., 2016) in which we simply marked the words $t_j$ of a machine-translated segment $T$ as *OK* (no changes are needed) or as *BAD* (needing editing). Now we also mark the gaps $\gamma_j$ after each word $t_j$ as *OK* (no insertions are needed) or as *BAD* (needing the insertion of one or more words). In addition, we use the edit operations predicted at the word level to estimate quality at the sentence level.

The paper is organized as follows: section 2 briefly reviews previous work on word-level MT QE; section 3 describes the method used to label words and gaps, paying special attention to the features extracted (sections 3.1 and 3.2) and the neural network (NN) architecture and its training (section 3.3); section 4 describes the datasets used; section 5 shows the main results; and, finally, section 6 closes the paper with concluding remarks.

## 2 Related work

Pioneering work on word-level MT QE dealt with predictive/interactive MT (Gandrabur and Foster, 2003; Blatz et al., 2004; Ueffing and Ney, 2005, 2007), often under the name of *confidence estimation*. Estimations relied on the internals of the actual MT system —for instance, studying the $n$-best translations (Ueffing and Ney, 2007)— or used external sources of bilingual information; for instance, both Blatz et al. (2004) and Ueffing and Ney (2005) used probabilistic dictionaries; in the case of Blatz et al. (2004), as one of many features in a binary classifier for each word.

The last decade has witnessed an explosion of work in word-level MT QE, with most of the recent advances made by participants in the shared tasks on MT QE at the different editions of the Conference on Statistical Machine Translation (WMT). Therefore, we briefly review those papers related to our approach: those using an external bilingual source such as an MT system and those using NN.

As regards work using *external bilingual resources*, we can highlight four groups of contributions:

- To estimate the sentence-level quality of MT output for a source segment $S$, Biçici (2013) chooses sentence pairs from a parallel corpus which are close to $S$, and builds an SMT system whose internals when translating $S$ are examined to extract features.
- MULTILIZER, one of the participants in the sentence-level MT QE task at WMT 2014 (Bojar et al., 2014) uses other MT systems to translate $S$ into the target language (TL) and $T$ into the source language (SL). The results are compared to the original SL and TL segments to obtain indicators of quality.
- Blain et al. (2017) use *bilexical embeddings* (obtained from SL and TL word embeddings

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 801–808
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64091

and word-aligned parallel corpora) to model the strength of the relationship between SL and TL words, in order to estimate sentence-level and word-level MT quality.

- Finally, Esplà-Gomis et al. (2015a,b), and Esplà-Gomis et al. (2016) perform word-level MT QE by using other MT systems to translate sub-segments of $S$ and $T$ and extracting features describing the way in which these translated sub-segments match sub-segments of $T$. This is the work most related to the one presented in this paper.

Only the last two groups of work actually tackle the problem of *word-level* MT QE, and none of them are able to identify the gaps where insertions are needed.

As regards the use of *neural networks* (NN) in MT QE, we can highlight a few contributions:

- Kreutzer et al. (2015) use a deep feed-forward NN to process the concatenated vector embeddings of neighbouring TL words and (word-aligned) SL words into feature vectors — extended with the baseline features provided by WMT15 (Bojar et al., 2015) organizers— to perform word-level MT QE.
- Martins et al. (2016) achieved the best results in the word-level MT QE shared task at WMT 2016 (Bojar et al., 2016) by combining a feed-forward NN with two recurrent NNs whose predictions were fed into a linear sequential model together with the baseline features provided by the organizers of the task. An extension (Martins et al., 2017) uses the output of an automatic post-editing tool, with a clear improvement in performance.
- Kim et al. (2017a,b) obtained in WMT 2017 (Bojar et al., 2017) results which were better or comparable to those by Martins et al. (2017), using a three-level stacked architecture trained in a multi-task fashion, combining a neural word prediction model trained on large-scale parallel corpora, and word- and sentence-level MT QE models.

Our approach uses a much simpler architecture than the last two approaches, containing no recurrent NNs, but just feed-forward NNs applied to a fixed-length context window around the word or gap about which a decision is being made (similarly to a convolutional approach). This makes our approach easier to train and parallelize.

## 3 Method

The approach presented here builds on previous work by the same authors (Esplà-Gomis et al., 2015a,b; Esplà-Gomis et al., 2016) in which insertion positions were not yet predicted and a slightly different feature set was used. As in the original papers, here we use black-box bilingual resources from the Internet. In particular, we use, for each language pair, the statistical MT phrase tables available at OPUS[1] to spot sub-segment correspondences between the SL segment $S$ and its machine translation $T$ into the TL (see section 4.2 for details). This is done by dividing both $S$ and $T$ into all possible (overlapping) sub-segments, or $n$-grams, up to a certain maximum length.[2] These sub-segments are then translated into the TL and the SL, respectively, by means of the phrase tables mentioned (lowercasing of sub-segments before and after translation is used to increase the chance of a match). These sub-segment correspondences are then used to extract several sets of features that are fed to a feed-forward NN in order to label the words and the gaps between words as *OK* or as *BAD*. One of the main advantages of this approach, when compared to the other approaches described below, is that it uses simple string-level bilingual information extracted from a publicly available source to build features that allow us to easily estimate quality for the words and inter-word gaps in $T$.

### 3.1 Features for word deletions

We define three sets of features to detect the words to be deleted: one taking advantage of the sub-segments $\tau$ that appear in $T$, $\text{Keep}_n(\cdot)$; another one that uses the translation frequency with which a sub-segment $\sigma$ in $S$ is translated as the sub-segment $\tau$ in $T$, $\text{Freq}_n^{\text{keep}}(\cdot)$; and a third one that uses the alignment information between $T$ and $\tau$ and which does not require $\tau$ to appear as a contiguous sub-segment in $T$, $\text{Align}_n^{\text{keep}}(\cdot)$.

**Features for word deletions based on sub-segment pair occurrences** (Keep) Given a set of sub-segment pairs $M = \{(\sigma, \tau)\}$ coming from the union of several phrase tables, the first set of features, $\text{Keep}_n(\cdot)$, is obtained by computing the amount of sub-segment translations $(\sigma, \tau) \in M$ with $|\tau| = n$ that confirm that word $t_j$ in $T$ should be kept in the translation of $S$. A sub-segment

[1] http://opus.nlpl.eu/

[2] For our submission, we used $L = 5$.

translation $(\sigma, \tau)$ confirms $t_j$ if $\sigma$ is a sub-segment of $S$, and $\tau$ is an $n$-word sub-segment of $T$ that covers position $j$. This set of features is defined as follows:

$$\text{Keep}_n(j, S, T, M) = \\ = \frac{|\{\tau : (\sigma, \tau) \in \text{conf}_n^{\text{keep}}(j, S, T, M)\}|}{|\{\tau : \tau \in \text{seg}_n(T) \wedge j \in \text{span}(\tau, T)\}|}$$

where $\text{seg}_n(X)$ represents the set of all possible $n$-word sub-segments of segment $X$, and function $\text{span}(\tau, T)$ returns the set of word positions spanned by the sub-segment $\tau$ in the segment $T$; if $\tau$ is found more than once in $T$, it returns all the possible positions spanned. Function $\text{conf}_n^{\text{keep}}(j, S, T, M)$ returns the collection of sub-segment pairs $(\sigma, \tau)$ that confirm a given word $t_j$, and is defined as:

$$\text{conf}_n^{\text{keep}}(j, S, T, M) = \\ = \{(\sigma, \tau) \in \text{match}_n(M, S, T) : j \in \text{span}(\tau, T)\}$$

where $\text{match}_n(M, S, T))$ is the set of phrase pairs in $M$ with $n$ words in the target that are found in the segment pair $(S, T)$, and where $\text{seg}_*(X)$ is similar to $\text{seg}_n(X)$ but without length constraints.[3]

**Features for word deletions based on sub-segment pair occurrences using translation frequency** ($\text{Freq}_n^{\text{keep}}$) The second set of features uses the probabilities of subsegment pairs. To obtain these probabilities from a set of phrase tables, we first use the count of joint occurrences of $(\sigma, \tau)$ provided in each phrase table. Then, when looking up a SL sub-segment $\sigma$, the probability $p(\tau|\sigma)$ is computed across all phrase tables from the accumulated counts. Finally, we define $\text{Freq}_n^{\text{keep}}(\cdot)$ as:

$$\text{Freq}_n^{\text{keep}}(j, S, T, M) = \\ = \sum_{(\sigma,\tau)\in\text{conf}_n^{\text{keep}}(j,S,T,M)} p(\tau|\sigma).$$

**Features for word deletions based on word alignments of partial matches** ($\text{Align}_n^{\text{keep}}$) The third set of features takes advantage of partial matches, that is, of sub-segment pairs $(\sigma, \tau)$ in which $\tau$ does not appear as such in $T$. This set of features is defined as:

$$\text{Align}_n^{\text{keep}}(j, S, T, M, e) = \\ = \sum_{\tau\in\text{segs\_edop}_n(j,S,T,M,e)} \frac{|\text{LCS}(\tau, T)|}{|\tau|} \quad (1)$$

where $\text{LCS}(X, Y)$ returns the word-based longest common sub-sequence between segments $X$ and $Y$, and $\text{segs\_edop}_n(j, S, T, M, e)$ returns the set of sub-segments $\tau$ of length $n$ from $M$ that are a translation of a sub-segment $\sigma$ from $S$ and in which, after computing the LCS with $T$, the $j$-th word $t_j$ is assigned the edit operation $e$:[4]

$$\text{segs\_edop}_n(j, S, T, M, e) = \\ = \{(\tau : (\sigma, \tau) \in M \wedge \sigma \in \text{seg}_*(S) \\ \wedge |\tau| = n \wedge \text{editop}(t_j, T, \tau) = e\} \quad (2)$$

where $\text{editop}(t_j, T, \tau)$ returns the edit operation assigned to $t_j$ and $e$ is either `delete` or `match`. If $e$ = `match` the resulting set of features provides evidence in favour of keeping the word $t_j$ unedited, whereas when $e$ = `delete` it provides evidence in favour of removing it. Note that features $\text{Align}_n^{\text{keep}}(\cdot)$ are the only ones to provide explicit evidence that a word should be deleted.

The three sets of features described so far, $\text{Keep}_n(\cdot)$, $\text{Freq}_n^{\text{keep}}(\cdot)$, and $\text{Align}_n^{\text{keep}}(\cdot)$, are computed for $t_j$ for all the values of sub-segment length $n \in [1, L]$. Features $\text{Keep}_n(\cdot)$ and $\text{Freq}_n^{\text{keep}}(\cdot)$ are computed by querying the collection of sub-segment pairs $M$ in both directions (SL–TL and TL–SL). Computing $\text{Align}_n^{\text{keep}}(\cdot)$ only queries $M$ in one direction (SL–TL) but is computed twice: once for the edit operation `match`, and once for the edit operation `delete`.

### 3.2 Features for insertion positions

In this section, we describe three sets of features —based on those described in section 3.1 for word deletions— designed to detect insertion positions. The main difference between them is that the former apply to words, while the latter apply to gaps; we will call $\gamma_j$ the gap after word $t_j$.[5]

**Features for insertion positions based on sub-segment pair occurrences** (NoInsert) The first set of features, $\text{NoInsert}_n(\cdot)$, based on the $\text{Keep}_n(\cdot)$ features for word deletions, is defined as follows:

$$\text{NoInsert}_n(j, S, T, M) = \\ \frac{|\{\tau : (\sigma, \tau) \in \text{conf}_n^{\text{noins}}(j, S, T, M)\}|}{|\{\tau : \tau \in \text{seg}_n(T) \wedge [j, j+1] \subseteq \text{span}(\tau, T)\}|}$$

[3] Esplà-Gomis et al. (2015a) conclude that constraining only the length of $\tau$ leads to better results than constraining both $\sigma$ and $\tau$.

[4] Note that the sequence of edit operations needed to transform $X$ in $Y$ is a by-product of computing $\text{LCS}(X, Y)$; these operations are `insert`, `delete` or `match` (when the corresponding word does not need to be edited).

[5] Note that the index of the first word in $T$ is 1, and gap $\gamma_0$ corresponds to the space before the first word in $T$.

where function $\mathrm{conf}_n^{\mathrm{noins}}(j, S, T, M)$ returns the collection of sub-segment pairs $(\sigma, \tau)$ covering a given gap $\gamma_j$, and is defined as:

$$\begin{aligned}&\mathrm{conf}_n^{\mathrm{noins}}(j, S, T, M) = \\ &\{(\sigma, \tau) \in \mathrm{match}_n(M, S, T) : \\ &\quad [j, j+1] \subseteq \mathrm{span}(\tau, T)\}\end{aligned}$$

$\mathrm{NoInsert}_n(\cdot)$ accounts for the number of times that the translation of sub-segment $\sigma$ from $S$ makes it possible to obtain a sub-segment $\tau$ that covers the gap $\gamma_j$, that is, a $\tau$ that covers both $t_j$ and $t_{j+1}$. If a word is missing in gap $\gamma_j$, one would expect to find fewer sub-segments $\tau$ that cover this gap, therefore obtaining low values for $\mathrm{NoInsert}_n(\cdot)$, while if there are no words missing in $\gamma_j$, one would expect more sub-segments $\tau$ to cover the gap, therefore obtaining values of $\mathrm{NoInsert}_n(\cdot)$ closer to 1. In order to be able to identify insertion positions before the first word or after the last word, we use imaginary sentence boundary words $t_0$ and $t_{|T|+1}$, which can also be matched,[6] thus allowing us to obtain evidence for gaps $\gamma_0$ and $\gamma_{|T|}$.

**Features for insertion positions based on sub-segment pair occurrences using translation frequency** ($\mathrm{Freq}_n^{\mathrm{noins}}$) Analogously to $\mathrm{Freq}_n^{\mathrm{keep}}(\cdot)$ above, we define the feature set $\mathrm{Freq}_n^{\mathrm{noins}}(\cdot)$, now for gaps:

$$\mathrm{Freq}_n^{\mathrm{noins}}(j, S, T, M) = \\ = \sum_{(\sigma,\tau)\in \mathrm{conf}_n^{\mathrm{noins}}(j,S,T,M)} p(\tau|\sigma)$$

**Features for insertion positions based on word alignments of partial matches** ($\mathrm{Align}_n^{\mathrm{noins}}$) Finally, the set of features $\mathrm{Align}_n^{\mathrm{keep}}(\cdot)$ for word deletions can be easily repurposed to detect the need for insertions by setting the edit operation $e$ in eq. (1) to `match` and `insert` and redefining eq. (2) as

$$\begin{aligned}\mathrm{segs\_edop}_n(j, S, T, M, e) = \{(\tau : (\sigma, \tau) \in M \\ \wedge |\tau| = n \\ \wedge \mathrm{editop}(t_j, \tau, T) = e\}\end{aligned}$$

where the LCS is computed between $\tau$ and $T$, rather than the other way round.[7] We shall refer to this last set of features for insertion positions as $\mathrm{Align}_n^{\mathrm{noins}}(\cdot)$.

The sets of features for insertion positions, $\mathrm{NoInsert}_n(\cdot)$, $\mathrm{Freq}_n^{\mathrm{noins}}(\cdot)$ and $\mathrm{Align}_n^{\mathrm{noins}}(\cdot)$, are computed for gap $\gamma_j$ for all the values of sub-segment length $n \in [2, L]$. As in the case of the feature sets employed to detect deletions, the first two sets are computed by querying the set of sub-segment pairs $M$ via the SL or via the TL, while the latter can only be computed by querying $M$ via the SL for the edit operations `insert` and `match`.

### 3.3 Neural network architecture and training

We use a two-hidden-layer feed-forward NN to jointly predict the labels (*OK* or *BAD*) for word $t_j$ and gap $\gamma_i$, using features computed at word positions $t_{i-C}, t_{i-C+1}, \ldots, t_{i-1}, t_i, t_{i+1}, \ldots, t_{i+C-1}$, $t_{i+C}$ and at gaps $\gamma_{i-C}$, $\gamma_{i-C+1}, \ldots$, $\gamma_{i-1}$, $\gamma_i$, $\gamma_{i+1}, \ldots$, $\gamma_{i+C-1}$, $\gamma_{i+C}$, where $C$ represents the amount of left and right context around the word and gap being predicted.

The NN architecture has a modular first layer with ReLU activation functions, in which the feature vectors for each word and gap, with $F$ and $G$ features respectively, are encoded into intermediate vector representations ("embedding") of the same size; word features are augmented with the baseline features provided by the organizers. The weights for this first layer are the same for all words and for all gaps (parameters are tied). A second layer of ReLU units combines these representations into a single representation of the same length $(2C + 1)(F + G)$. Finally, two sigmoid neurons in the output indicate, respectively, if word $t_i$ has to be tagged as *BAD*, or if gap $\gamma_i$ should be labelled as *BAD*. Preliminary experiments confirmed that predicting word and gap labels with the same NN lead to better results than using two independent NNs.

The output of each of the sigmoid output units is additionally independently thresholded (Lipton et al., 2014) using a line search to establish thresholds that optimize the product of the $F_1$ score for *OK* and *BAD* categories on the development sets. This is done since the product of the $F_1$ scores is the main metric of comparison of the shared task, but it cannot be directly used as the objective function of the training as it is not differentiable.

Training was carried out using the Adam stochastic gradient descent algorithm to optimize cross-entropy. A dropout regularization of 20% was applied on each hidden layer. Training was stopped when results on the development set did not improve for 10 epochs. In addition, each network was trained 10 times with different uniform initializa-

[6] These boundary words are annotated in $M$ when this resource is built.

[7] It is worth noting that $\mathrm{LCS}(X, Y) = \mathrm{LCS}(Y, X)$, but the sequences of edit operations obtained as a by-product are different in each case.

tions (He et al., 2015), choosing the parameter set performing best on the development set.

Preliminary experiments have led us to choose a value $C = 3$ for the number of words and gaps both to the left and to the right of the word and gap for which a prediction is being made; smaller values such a $C = 1$ gave, however, a very similar performance.

## 4 Experimental setting

### 4.1 Datasets provided by the organizers

Six datasets were provided for the shared task on MT QE at WMT 2018 (Specia et al., 2018), covering four language pairs —English–German (EN–DE), German–English (DE–EN), English–Latvian (EN–LV), and English–Czech (EN–CS)— and two MT systems —statistical MT (SMT) and neural MT (NMT). Each dataset is split into training, development and test sets. From the data provided by the organizers of the shared task, the approach in this paper used:

1. set of segments $S$ in source language,
2. set of translations $T$ of the SL segment produced by an MT system,
3. word-level MT QE gold predictions for each word and gap in each translation $T$, and
4. baseline features[8] for word-level MT QE.

Regarding the baseline features, the organizers provided 28 features per word in the dataset, from which we only used the 14 numeric features plus the part-of-speech category (one-hot encoded). This was done for the sake of simplicity of our architecture. It is worth mentioning that no valid baseline features were provided for the EN–LV datasets. In addition, the large number of part-of-speech categories in the EN–CS dataset led us to discard this feature in this case. As a result, 121 baseline features were obtained for EN–DE (SMT), 122 for EN–DE (NMT), 123 for DE-EN (SMT), 14 for EN–CS (SMT), and 0 for EN–LV (SMT) and EN–LV (NMT).

### 4.2 External bilingual resources

As described above, our approach uses ready-made, publicly available phrase tables as bilingual resources. In particular, we have used the cleaned phrase tables available on June 6, 2018 in OPUS for the language pairs involved. These phrase tables were built on a corpus of about 82 million pairs of sentences for DE–EN, 7 million for EN–LV, and 61 million for EN–CS. Phrase tables were available only for one translation direction and some of them had to be inverted (for example, in the case of EN–DE or EN–CS).

## 5 Results

This section describes the results obtained by the UAlacant system in the MT QE shared task at WMT 2018 (Specia et al., 2018), which are reported in Table 1. Our team participated in two sub-tasks: sentence-level MT QE (task 1) and word-level MT QE (task 2). For sentence-level MT QE we computed the number of word-level operations predicted by our word-level MT QE approach and normalized it by the length of each segment $T$, in order to obtain a metric similar to TER. The words tagged as *BAD* followed by gaps tagged as *BAD* were counted as replacements, the rest of words tagged as *BAD* were counted as deletions, and the rest of gaps tagged as *BAD* were counted as one-word insertions.[9] This metric was used to participate both in the scoring and ranking sub-tasks.

Columns 2 to 5 of Table 1 show the results obtained for task 1 in terms of the Pearson's correlation $r$ between predictions and actual HTER, mean average error (MAE), and root mean squared error (RMSE), as well as Sperman's correlation $\rho$ for ranking.

Columns 6 to 11 show the results for task 2 in terms of $F_1$ score both for categories *OK* and *BAD*,[10] together with the product of both $F_1$ scores, which is the main metric of comparison of the task. The first three columns contain the results for the sub-task of labelling words while the last three columns 9 to 11 contain the results for the sub-task of labelling gaps.

As can be seen, the best results were obtained for the language pair DE–EN (SMT). Surprisingly, the results obtained for EN–LV (NMT) were also specially high for word-level and sentence-level MT QE. These results for the latter language pair are unexpected for two reasons: first, because no baseline features were available for word-level MT

[8] https://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox_baseline_17

[9] Note that this approach is rather limited, as it ignores block shifts and the number of words to be inserted in a gap, which are basic operations to compute the actual TER value.

[10] For word deletion identification, a word marked as *BAD* means that the word needs to be deleted, while in the case of insertion position identification, if a gap is marked as *BAD* it means that one or more words need to be inserted there.

| Dataset | sentence-level |  |  |  | word-level (words) |  |  | word-level (gaps) |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
|  | $r$ | **MAE** | **RMSE** | $\rho$ | $F_{\text{BAD}}$ | $F_{\text{OK}}$ | $F_{\text{MULTI}}$ | $F_{\text{BAD}}$ | $F_{\text{OK}}$ | $F_{\text{MULTI}}$ |
| EN–DE SMT | 0.45 | 0.15 | 0.19 | 0.44 | 0.35 | 0.81 | 0.29 | 0.33 | 0.96 | 0.32 |
| EN–DE NMT | 0.35 | 0.14 | 0.20 | 0.41 | 0.22 | 0.86 | 0.19 | 0.12 | 0.98 | 0.12 |
| DE–EN SMT | 0.63 | 0.12 | 0.17 | 0.60 | 0.43 | 0.87 | 0.37 | 0.33 | 0.97 | 0.32 |
| EN–LV SMT | 0.36 | 0.20 | 0.26 | 0.34 | 0.27 | 0.82 | 0.22 | 0.15 | 0.94 | 0.14 |
| EN–LV NMT | 0.56 | 0.17 | 0.22 | 0.55 | 0.44 | 0.80 | 0.36 | 0.17 | 0.95 | 0.16 |
| EN–CS SMT | 0.43 | 0.18 | 0.23 | 0.46 | 0.42 | 0.75 | 0.31 | 0.15 | 0.95 | 0.15 |

Table 1: Results for sentence-level MT QE (columns 2–5) in terms of the Pearson's correlation $r$, MAE, RMSE, and Sperman's correlation $\rho$ (for ranking). Results for the task of word labelling (columns 6–8) and gap labelling (columns 9–11) in terms of the $F_1$ score for class *BAD* ($F_{\text{BAD}}$), the $F_1$ score for class *OK* ($F_{\text{OK}}$) and the product of both ($F_{\text{MULTI}}$).

QE task for this language pair, and second, because the size of the parallel corpora from which phrase tables for this language pair were extracted were an order of magnitude smaller. One may think that the coverage of machine translation by the phrase tables could have an impact on these results. To confirm this, we checked the fraction of words in each test set that were not covered by any subsegment pair $(\sigma, \tau)$. This fraction ranges from 15% to 4% depending on the test set, and has the lowest value for EN–LV (NMT); however, it is not clear that a higher coverage always leads to a better performance as one of the datasets with a better coverage was EN–LV (SMT) (5%) which, in fact, obtained the worst results in our experiments.

It is worth noting that, when looking at the results obtained by other participants, the differences in performance between the different datasets seems to be rather constant, showing, for example, a drop in performance for EN–DE (NMT) and EN–LV (SMT); this lead us to think that the test set might be more difficult in these cases. One thing that we could confirm is that, for these two datasets, the ratio of *OK*/*BAD* samples for word-level MT QE is lower, which may make the classification task more difficult.

In comparison with the rest of systems participating in this task, UAlacant was the best-performing one in the sub-task of labelling gaps for 3 out of the 6 datasets provided (DE–EN SMT, EN–LV SMT, and EN–LV NMT). Results obtained for the subtask of labelling words were poorer and usually in the lower part of the classification. However, the sentence-level MT QE submissions, which build on the labels predicted for words and gaps by the word-level MT QE system, performed substantially better and outperformed the baseline for all the datasets but EN–DE (NMT) and, for EN–LV (NMT), it even ranked third.

As said above, one of the main advantages of this approach is that it can be trained with limited computational resources. In our case, we trained our systems on a AMD Opteron(tm) Processor 6128 CPU with 16 cores and, for the largest set of features (dataset DE–EN SMT), training took 2,5 hours, about 4 minutes per epoch.[11]

## 6 Concluding remarks

We have presented a simple MT word-level QE method that matches the content of publicly available statistical MT phrase pairs to the source segment $S$ and its machine translation $T$ to produce a number of features at each word and gap. To predict if the current word has to be deleted or if words have to be inserted in the current gap, the features for the current word and gap and $C$ words and gaps to the left and to the right are processed by a two-hidden-layer feed-forward NN. When compared with other participants in the WMT 2018 shared task, our system ranks first in labelling gaps for 3 of the 6 language pairs, but does not perform too well in labelling words. We also used word-level estimations to approximate TER. We participated with this approximation in the sentence-level MT QE sub-task obtaining a reasonable performance ranking, for almost all datasets, above the baseline.

One of the main advantages of the work presented here is that it does not require huge computational resources, and it can be trained even on a CPU in a reasonable time.

## Acknowledgments

Work funded by the Spanish Government through the EFFORTUNE project (project number TIN-2015-69632-R).

[11] Total training time corresponds to 35 epochs.

## References

Ergun Biçici. 2013. Referential translation machines for quality estimation. In *Proceedings of the 8th Workshop on Statistical Machine Translation*, pages 343–351, Sofia, Bulgaria.

Frédéric Blain, Carolina Scarton, and Lucia Specia. 2017. Bilexical embeddings for quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 545–550.

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pages 315–321, Geneva, Switzerland.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, MD, USA.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal.

Miquel Esplà-Gomis, Felipe Sánchez-Martínez, and Mikel L. Forcada. 2015a. UAlacant word-level machine translation quality estimation system at WMT 2015. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 309–315, Lisbon, Portugal.

Miquel Esplà-Gomis, Felipe Sánchez-Martínez, and Mikel L. Forcada. 2015b. Using on-line available sources of bilingual information for word-level machine translation quality estimation. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, pages 19–26, Antalya, Turkey.

Miquel Esplà-Gomis, Felipe Sánchez-Martínez, and Mikel L. Forcada. 2016. UAlacant word-level and phrase-level machine translation quality estimation systems at WMT 2016. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 782–786.

Simona Gandrabur and George Foster. 2003. Confidence estimation for translation prediction. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 95–102, Edmonton, Canada.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1026–1034, Washington, DC, USA.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017a. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(1):3.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017b. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568.

Julia Kreutzer, Shigehiko Schamoni, and Stefan Riezler. 2015. Quality estimation from ScraTCH (QUETCH): Deep learning for word-level translation quality estimation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 316–322, Lisbon, Portugal. Association for Computational Linguistics.

Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. 2014. Optimal thresholding of classifiers to maximize F1 measure. In *Machine Learning and Knowledge Discovery in Databases*, pages 225–239, Berlin, Heidelberg. Springer Berlin Heidelberg.

André F. T. Martins, Ramón Astudillo, Chris Hokamp, and Fabio Kepler. 2016. Unbabel's participation in the WMT16 word-level translation quality estimation shared task. In *Proceedings of the First Conference on Machine Translation*, pages 806–811, Berlin, Germany.

André FT Martins, Marcin Junczys-Dowmunt, Fabio N Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics*, 5:205–218.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón F. Astudillo, and André Martins. 2018. Findings of the WMT 2018 Shared Task on Quality Estimation. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium.

Nicola Ueffing and Hermann Ney. 2005. Application of word-level confidence measures in interactive statistical machine translation. In *Proceedings of the 10th European Association for Machine Translation Conference "Practical applications of machine translation"*, pages 262–270, Budapest, Hungary.

Nicola Ueffing and Hermann Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40.

# Alibaba Submission for WMT18 Quality Estimation Task

**Jiayi Wang***, **Kai Fan***, **Bo Li, Fengming Zhou, Boxing Chen, Yangbin Shi, Luo Si**
Machine Intelligence Technology Lab, Alibaba Group
Hangzhou, China
{joanne.wjy, k.fan, shiji.lb, zfm104435, boxing.cbx, taiwu.syb, luo.si}@alibaba-inc.com

## Abstract

The goal of WMT 2018 Shared Task on Translation Quality Estimation is to investigate automatic methods for estimating the quality of machine translation results without reference translations. This paper presents the QE Brain system, which proposes the neural Bilingual Expert model as a feature extractor based on conditional target language model with a bidirectional transformer and then processes the semantic representations of source and the translation output with a Bi-LSTM predictive model for automatic quality estimation. The system has been applied to the sentence-level scoring and ranking tasks as well as the word-level tasks for finding errors for each word in translations. An extensive set of experimental results have shown that our system outperformed the best results in WMT 2017 Quality Estimation tasks and obtained top results in WMT 2018.

## 1 Introduction

Quality Estimation (QE) is a task to estimate the quality of a Machine Translation (MT) system without the presence of any manually annotated reference translations. It can serve in a variety of computer-aided scenarios such as translation results screening before release or translation quality comparison between different MT systems. Currently, the classical and widely-used method to evaluate an MT system is measured by BLEU (Papineni et al., 2002), a statistical language-independent metric that requires human golden references for validation. What if we expect to efficiently get the detailed quality evaluation feedbacks (e.g. sentence or token-wise scoring) from an extremely large number of machine translation outputs? An automatic method with no access to any reference is highly appreciated.

The common approach to automatic translation quality estimation is to transform the problem into a supervised regression or classification task for sentence-level scoring and word-level labeling respectively. Traditional baseline models in WMT 12-17 have two modules: human-crafted rule-based feature extraction model via QuEst++ (Specia et al., 2015) (sentence-level task) or Marmot[1] (word-level task); and an SVM regression with an RBF kernel as well as grid search algorithms for predicting how much effort is needed to fix translations to acceptable results (sentence-level task) or a sequence-labeling model with CRFSuit toolkit to predict which word in the translation output needs to be edited (word-level task). A recently proposed predictor-estimator model with stack propagation (Kim et al., 2017) is a recurrent neural network (RNN) based feature extractor and quality prediction model that ranked first place in WMT17. Another novel method is to train an Automatic Post-Editing (APE) system and adapt it to predict sentence-level quality scores and word-level quality labels (Martins et al., 2017). A promising APE system can serve as a guidance to QE system by explicitly explaining errors in the translation output.

Our submitted system for sentence and word level QE tasks in WMT18, named ***QE Brain*** has two phases: feature extraction and quality estimation. In the phase of feature extraction, it extracts high-level latent joint semantics and alignment information between the source and the translation output, relying on the "neural Bilingual Expert model" introduced by Fan et al. (2018) as a prior knowledge model, which is trained on a large parallel corpus. The high-level latent semantic features and manually designed mis-matching features (Fan et al., 2018) exported from the prior

* indicates equal contribution.

[1] https://github.com/qe-team/marmot

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 809–815
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64092

knowledge model are fed into a predictive model in the phase of quality estimation, with which the scoring prediction for the sentence-level task and erroneous or missing word predictions for the word-level task are targeted. This paper presents our submissions for the WMT18 Quality Estimation English-German and German-English Shared Tasks, namely, (i) a sentence-level QE scoring prediction system and (ii) a word-level QE labeling prediction system including word predictions and gap predictions. Since both systems are supposed to understand the complex semantic relationship between the source and the translation output, the features produced by a pre-trained neural Bilingual Expert model can be shared by the two level tasks per language direction.

In Section 3, we will discuss several techniques to boost our system's performance. We make use of extra human-crafted baseline features including basic descriptive statistics, language model (LM) probabilities and alignments information of the source and the translation output. They are combined with features from the neural Bilingual Expert model to predict the sentence-level scores. In addition, to make up the shortage of QE training data, we apply the round-trip translation technique to generate some artificial QE data that increases the error diversity and prevents overfitting. To further enhance our model's performance, we use a greedy algorithm based ensemble selection method to decrease the individual error among a bunch of single quality estimation models.

## 2 QE Brain Baseline Model

QE Brain base single model contains a feature extractor and a quality estimator. The feature extractor relies on the Bilingual Expert model to extract features representing latent semantic information of the source and translation pair. These features will be fed into a quality estimator to estimate the translation quality.

The Bilingual Expert model uses self-attention mechanism and transformer neural networks to construct a bidirectional transformer architecture (Fan et al., 2018), serving as a conditional language model. It is used to predict every single word in the target sentence given the entire source sentence and its context . The Bilingual Expert model consists of three modules: (i) transformer self-attention based encoder for the source sentence, (ii) forward and backward encoders for the target sentence with the masked self-attention in the transformer decoder module, (iii) reconstruction for the target sentence. Once the model is fully trained, we can use the prior knowledge learned from the Bilingual Expert model to extract the features for the subsequent translation quality estimator. There are two kinds of features upon the Bilingual Expert model defined by Fan et al. (2018): model derived features of latent representations and manually extracted mismatching features.

When we perform quality estimation on a source and translation pair, we need to obtain the semantics information of the source and the translation output and their alignment information. We can assume that it is more likely for the model to predict a correct target word if only few words around it are incorrect. Fan et al. (2018) claims that both the latent representations of the $k$-th word in the translation output and its mismatching features that reflect the error severity if it is a mistake are sufficiently beneficial to the downstream quality predictive model. Choices of the quality estimation models are compared as well. It is found that the bi-directional LSTM (Graves and Schmidhuber, 2005) will be appropriate in the QE situation. We treat the feature extraction model based on the neural Bilingual Expert model and the quality estimation based on Bi-LSTM model as our baseline system.

## 3 Boosting the QE Model Performance

### 3.1 Human-crafted Features

Along with the features produced by the Bilingual Expert model, we extract another 17 QE baseline features for the sentence-level task using QuEst++ and additional resources (source and target corpora, language models, ngram counts and lexical translation tables) provided on the WMT18 QE website[2]. Kozlova et al. (2016) verifies the significance of these features using Random Forest (Breiman, 2001). Four of them are the most crucial among all according to their degrees of importance.

- percentage of trigrams in quartile 4 of frequency of source words in a corpus of the source language
- LM probability of source sentence

[2] http://www.statmt.org/wmt18/quality-estimation-task.html

- percentage of bigrams in quartile 4 of frequency of source words in a corpus of the source language
- average number of translations per source word in the sentence

Language models (LM) assign probabilities to generate hypotheses in the target language informing lexical selection in statistical machine translation (SMT). It is reasonable that three of the above four baseline features are derived from the LM. Moreover, alignment models can essentially help SMTs determine translational correspondences between the N-grams in the source with those of the same meanings in the target. Particularly, a satisfying translation result can contain as many translated words as possible, according to an alignment model, IBM model 1 or 2. Consequently, average number of translations per source word in the sentence becomes large.

Fan et al. (2018) proposed to use the concatenation of the model derived and mis-matching features as input of a Bi-LSTM quality predictive model. The sentence-level score prediction can be formulated as a regression problem with the objective function,

$$\arg\min \left\| h - \text{sigmoid}\left(\mathbf{w}^\top [\overrightarrow{\mathbf{h}_T}; \overleftarrow{\mathbf{h}_T}]\right)\right\|_2^2 \tag{1}$$

where $\overrightarrow{\mathbf{h}_T}$ and $\overleftarrow{\mathbf{h}_T}$ are the hidden states of the last time stamps of the Bi-LSTM's output, $h$ represents the translation score (HTER) and $\mathbf{w}$ is a vector. Alternatively, we introduce the human-crafted features as additional linear components for the predictive layer with a sigmoid activation function. Therefore, the objective function can be rewritten as,

$$\arg\min \left\| h - \text{sigmoid}\left(\mathbf{w}^\top [\overrightarrow{\mathbf{h}_T}; \overleftarrow{\mathbf{h}_T}; \mathbf{f}_h]\right)\right\|_2^2 \tag{2}$$

where $\mathbf{f}_h$ is the 17-dimensional QE baseline features.

### 3.2 Artificial QE Data Construction

Unlike stacking of an APE-based QE system and a "pure" QE system trained only on the provided QE training dataset (Martins et al., 2017), we came up with the idea to take advantage of the artificial training data augmentation technique (Junczys-Dowmunt and Grundkiewicz, 2016) in the APE task to provide more supplementary training data, aiming to increase the diversity of erroneous translations during the training process so that it can reduce the overfitting of our model. We trained two English-German quality estimation models with (i) the real QE training data alone or (ii) the real and artificial QE data, and evaluated them on the development data and the data made up with 1800 random samples from the real QE training data to investigate the robustness of them. As shown in Fig 1, the model trained with (ii) (Experiment 2) is more robust than the model trained with (i) (Experiment 1), but can achieve comparable performance on the development data.

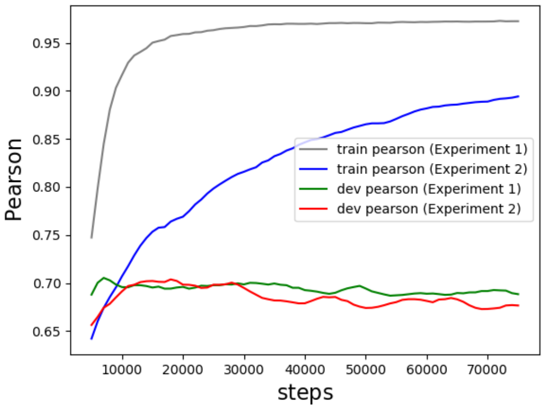


Figure 1: Robustness analysis on English-German QE model. Experiment 1: model trained with real QE data; Experiment 2: model trained with real and artificial QE data

The round-trip translation process can produce literal translations that may be similar to post-edited triplets including sources (SRC), translation outputs (MT) and post editions (PE). In order to mimic the QE data, we randomly pick triplets generated by the round-trip translation technique according to the distribution of HTERs in the real QE training and development data.

### 3.3 Greedy Ensemble Selection

To generate an ensemble of submissions for the WMT 18 QE task, the simplest methods are averaging the predictive scores for the sentence level and majority voting for the predictive labels for the word level from a number of single models. Homogeneous models can be derived from performing the same learning methodology but with different hyper-parameters of the model architecture including the neural Bilingual Expert model and Bi-LSTM quality predictive model.

In the sentence level, adding human-crafted features can be optional when we make different assumptions about the features of source and translation pairs. Under this situation, heterogeneous models can be derived from performing the same learning algorithm on different datasets. We can also use the Byte-Pair Encoding (BPE) tokenization as a substitution for a word tokenization in text pre-processing. Fan et al. (2018) compared the performance of the word and BPE tokenization on both sentence and word levels in WMT 18 and the results show that the models with BPE tokenization can produce comparable or better results than those with word tokenization.

In general, the ensemble output of $K$ single models can be produced by the following objective function,

$$\arg\max_{t_k} \sum_{k=1}^{K} w_k m_k \left(X = x, T = t_k\right) \tag{3}$$

where $m_k$ is the $k$-th single model that has probability distribution $m_k\left(x, t_k\right)$ with its corresponding weight $w_k$. $X$ represents the feature instance of a single model, and $T$ represents the HTER or the word label where $t_k$ can be a continuous quality score or an OK/BAD label respectively. We assign equal weights to every single model in our case for simplicity.

Since not every single model in the ensemble is always needed for the optimized prediction, it is appropriate to select a subset from all candidate models. We follow the greedy ensemble selection algorithm, Focused Ensemble Selection (FES ) (Partalas et al., 2008), to reduce the size of averaging ensembles but improve its efficiency and predictive performance.

In the sentence level, FES's output is averaging HTER scores of selected single models. However, in the word level, the ensemble can be made by majority voting of the binary predictions for selected single models or averaging their probabilities of predicting the word as OK. We use the development data for evaluation under the assumption that the development data and the test data are from the same distribution, even if it might be susceptible to overfitting. However, we did not observe this phenomena in results released for the test data in WMT18 QE task.

# 4 Experiments

## 4.1 Experimental Settings

### 4.1.1 Data for Bilingual Expert Model

We evaluated our system, QE Brain, for the WMT17/18 QE task for sentence/word-level in English-German and German-English. The followings are data resources that we used for training the neural Bilingual Expert model,

- parallel corpora released for the WMT17/18 News Machine Translation Task[3]
- UFAL Medical Corpus and Khresmoi development data release for the WMT18 Biomedical Translation Task[4]
- source and target corpora MT training data released in the additional resources for the WMT18 QE Task
- src-pe pairs for for the WMT17/18 QE Task

We filtered all the corpora except src-pe pairs with basic rules to guarantee the quality. A "high-quality" sentence pair should both start with a Unicode letter character, the lengths of them are equal to or less than 70, and the length ratio of the source sentence and the target one should be bounded by 1/3 and 3. The total resulting qualifying parallel corpora roughly include 13 million for WMT17 QE tasks and 29 million for WMT18 QE tasks.

### 4.1.2 Data for Quality Estimation Model

The data for quality estimation contains two parts: (i) real QE data provided by WMT QE organizers; (ii) artificial QE data generated by the round-trip translation technique (Junczys-Dowmunt and Grundkiewicz, 2016). We first combined the real QE data with the artificial QE data to train a baseline quality estimation model, then fine tuned the model with the real QE data alone. The English-German IT domain artificial QE data can be obtained directly from the additional resources of WMT18 Auto Post-Editing task[5] created by Junczys-Dowmunt and Grundkiewicz (2016). We applied the English-German artificial QE data on

[3] http://www.statmt.org/wmt18/translation-task.html

[4] http://www.statmt.org/wmt18/biomedical-translation-task.html

[5] http://www.statmt.org/wmt18/ape-task.html

|  | test 2017 en-de |  |  |  |  | test 2017 de-en |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | **Pearson's** $r \uparrow$ | MAE $\downarrow$ | RMSE $\downarrow$ | **Spearman's** $\rho \uparrow$ | DeltaAvg $\uparrow$ | **Pearson's** $r \uparrow$ | MAE $\downarrow$ | RMSE $\downarrow$ | **Spearman's** $\rho \uparrow$ | DeltaAvg $\uparrow$ |
| Baseline | 0.397 | 0.136 | 0.175 | 0.425 | 0.0745 | 0.441 | 0.128 | 0.175 | 0.45 | 0.0681 |
| Unbabel | 0.641 | 0.128 | 0.169 | 0.652 | 0.1136 | 0.626 | 0.121 | 0.179 | 0.61 | 0.974 |
| POSTECH Single-Ensemble | 0.6731 | 0.1067 | 0.1412 | 0.7029 | 0.1198 | 0.7146 | 0.0942 | 0.1359 | 0.6327 | 0.1044 |
| POSTECH Multi-Ensemble | 0.6954 | 0.1019 | 0.1371 | 0.7253 | 0.1232 | 0.7280 | 0.0911 | 0.1332 | 0.6542 | 0.1064 |
| QE Brain Base Single Model | 0.6837 | 0.1001 | 0.1441 | 0.7091 | 0.1200 | 0.7099 | 0.0927 | 0.1394 | 0.6424 | 0.1018 |
| + HF | 0.6842 | 0.1013 | 0.1449 | 0.7150 | 0.1213 | 0.7085 | 0.0901 | 0.1406 | 0.6551 | 0.1040 |
| + FT | 0.6957 | 0.1001 | 0.1420 | 0.7205 | 0.1208 | 0.7128 | 0.0933 | 0.1394 | 0.6422 | 0.1013 |
| + HF/FT | 0.6813 | 0.1021 | 0.1460 | 0.7070 | 0.1197 | 0.7149 | 0.0889 | 0.1385 | 0.6596 | 0.1026 |
| QE Brain Ensemble | **0.7159** | 0.0965 | 0.1384 | **0.7402** | 0.1247 | **0.7338** | 0.0882 | 0.1333 | **0.6700** | 0.105 |

Table 1: Results of sentence-level scoring and ranking on WMT17. HF: human features; FT: fine-tune strategy with artificial QE data.

the SMT QE task. For the neural machine translation (NMT) QE task, we followed the same procedure but trained two NMT models (German-English and English-German) instead.

Similarly, when generating German-English Pharmacy domain artificial QE data, we first applied domain data selection to the English monolingual corpus admissible for the WMT18 News and Biomedical Translation data with cross-entropy filtering method and seed data set – post-editing training data and the English biomedical data. In total, we got 5 million domain-like sentences for the round-trip translation. Afterwards, we created two phrase-based translation models, English-German and German-English, using the parallel bilingual corpora for the WMT18 News and Biomedial Translation tasks but with different language models. The 5 million domain-like sentences as PEs would be first translated to German as SRCs and the SRCs would be then translated to English as MTs. Finally, we would have 5 million artificial APE training data, leading to 5 million artificial QE training data with corresponding HTERs and word labels via the TER tool.

We filtered the English-German and German-English artificial QE data according to the HTER distribution of the combination of QE training and development data, and randomly pick 300,000 triplets per language pair.

|  | **Pearson's** $r \uparrow$ | MAE $\downarrow$ | RMSE $\downarrow$ | **Spearman's** $\rho \uparrow$ |
|---|---|---|---|---|
| Method | test 2018 en-de SMT |  |  |  |
| Baseline | 0.3653 | 0.1402 | 0.1772 | 0.3809 |
| UNQE | 0.7000 | 0.0962 | 0.1382 | 0.7244 |
| QE Brain Ensemble 1 | 0.7308 | 0.0953 | 0.1383 | 0.7470 |
| QE Brain Ensemble 2 | **0.7397** | 0.0937 | 0.1362 | **0.7543** |
| Method | test 2018 en-de NMT |  |  |  |
| Baseline | 0.2874 | 0.1286 | 0.1886 | 0.4195 |
| UNQE | **0.5129** | 0.1114 | 0.1749 | **0.6052** |
| QE Brain Ensemble 1 | 0.5005 | 0.1134 | 0.1734 | 0.6002 |
| QE Brain Ensemble 2 | 0.5012 | 0.1131 | 0.1742 | 0.6049 |
| Method | test 2018 de-en SMT |  |  |  |
| Baseline | 0.3323 | 0.1508 | 0.1928 | 0.3247 |
| UNQE | **0.7667** | 0.0945 | 0.1315 | 0.7261 |
| QE Brain Ensemble 1 | 0.7539 | 0.0981 | 0.1355 | 0.7222 |
| QE Brain Ensemble 2 | 0.7631 | 0.0962 | 0.1328 | **0.7318** |

Table 2: Results of sent level QE on WMT2018

|  | F1-BAD | F1-OK | **F1-Multi** |
|---|---|---|---|
| Method | test 2017 en-de |  |  |
| Baseline | 0.407 | 0.886 | 0.361 |
| DCU | 0.614 | 0.910 | 0.559 |
| Unbabel | 0.625 | 0.906 | 0.566 |
| POSTECH Ensemble | 0.628 | 0.904 | 0.568 |
| QE Brain Base Single Model | 0.6407 | 0.9045 | 0.5795 |
| + FT | 0.6410 | 0.9083 | 0.5826 |
| QE Brain Ensemble | 0.6616 | 0.9128 | **0.6039** |
| Method | test 2017 de-en |  |  |
| Baseline | 0.365 | 0.939 | 0.342 |
| POSTECH Single-Ensemble | 0.552 | 0.936 | 0.516 |
| Unbabel | 0.562 | 0.941 | 0.529 |
| POSTECH Multi-Ensemble | 0.569 | 0.940 | 0.535 |
| QE Brain Base Single Model | 0.5750 | 0.9471 | 0.5446 |
| + FT | 0.5816 | 0.9470 | 0.5507 |
| QE Brain Ensemble | 0.5924 | 0.9475 | **0.5613** |
| Method | test 2018 en-de SMT |  |  |
| Baseline | 0.4115 | 0.8821 | 0.3630 |
| SHEF-PT | 0.5080 | 0.8460 | 0.4298 |
| QE Brain Ensemble 1 | 0.6616 | 0.9168 | 0.6066 |
| QE Brain Ensemble 2 | 0.6808 | 0.9175 | **0.6246** |
| Method | test 2018 en-de NMT |  |  |
| Baseline | 0.1973 | 0.9184 | 0.1812 |
| SHEF-PT | 0.3353 | 0.8691 | 0.2914 |
| QE Brain Ensemble 1 | 0.4750 | 0.9152 | **0.4361** |
| QE Brain Ensemble 2 | 0.4767 | 0.9149 | 0.4347 |
| Method | test 2018 de-en SMT |  |  |
| Baseline | 0.4850 | 0.9015 | 0.4373 |
| SHEF-PT | 0.4853 | 0.8741 | 0.4242 |
| QE Brain Ensemble 1 | 0.6475 | 0.9162 | 0.5932 |
| QE Brain Ensemble 2 | 0.6523 | 0.9217 | **0.6012** |

Table 3: Results of word-level word prediction on WMT17/18

| Method | F1-BAD | F1-OK | **F1-Multi** |
|---|---|---|---|
| UAlacante SBI | 0.1997 | 0.9444 | 0.1886 |
| SHEF-bRNN | 0.2710 | 0.9552 | 0.2589 |
| SHEF-PT | 0.2937 | 0.9618 | 0.2824 |
| QE Brain | 0.5109 | 0.9783 | **0.4999** |

Table 4: Results of word-level gap prediction on WMT18 En-De SMT

### 4.1.3 Model Settings

The number of layers for the self-attention encoder and forward/backward self-attention decoder are all set as 2, where we use 8-head self-attention in practice. The number of hidden units for feed-forward sub-layer is 512. The bilingual expert model is trained on 8 Nvidia P-100 GPUs for about 3 days until convergence. For translation QE model, we use only one layer Bi-LSTM, and it is trained on a single GPU. Notice that for the QE task of WMT17, it is prohibited to use any data from 2018, since the training data of 2018 includes some test data of 2017. The same setting is applied to all following experiments associated with 2017. We tuned all the hyper-parameters of our model on the development dataset to obtain the best single model, and report the corresponding results for test data.

We increased the model diversity from two perspectives. First, in terms of data resources, we experienced with three strategies: word/BPE tokenization, w/ or w/o artificial QE data and w/ or w/o human-crafted features for the sentence-level task. Secondly, we tuned the number of units for Bi-LSTM with 96 or 128 and training batch size with 32 or 64 from the model's perspective.

### 4.2 Evaluation Results

In this section, we will report the experimental results of our approach for WMT 2017 and 2018. For WMT17 QE task, we tried to verify our proposed strategies. For WMT18 QE task, we mainly participated in the sentence-level scoring and ranking tasks and the word-level word prediction tasks for English-German SMT, English-German NMT and German-English SMT. In addition, we also submitted results for the word-level gap predictions for English-German SMT. In Table 2, part of Table 3 and Table 4, results of WMT18 QE tasks are listed according to the WMT18 QE website.

#### 4.2.1 Ablation Study on WMT17 QE Task

Since we can access the translation outputs of human post-editing for test data, it provides an ideal held-out test data to verify our proposed strategies. We illustrated our results in Table 1 and part of Table 3 on WMT17 QE Task. The competitors are POSTECH, DCU and Unbabel. Their results can be found in (Bojar et al., 2017) , Section 4.4 and Section 4.5. We also listed the WMT QE baseline results for reference. The QE Brain base single model follows the exact training scheme in (Fan et al., 2018) with model derived features and mismatching features. In sentence level, either incorporating human features or the use of artificial QE data will positively contribute to the metrics. For Pearson's $r$, the single fine-tuning strategy yields the improvement $+0.01$ on English-German and $+0.003$ on German-English. For Spearman's $\rho$, the single model with human features improves the performance by $+0.006$ in English-German and $+0.013$ in German-English.

In word level, we did not use any human features, but we found fine-tune strategy can always improve the performance. For F1-Multi, the single fine-tuning strategy yields the improvement $+0.003$ on English-German and $+0.006$ on German-English. In general, with all these strategies, our single models can be comparable or better than the state-of-the-art (SOTA) ensemble systems of WMT17 QE task. Our ensemble models significantly outperform all of the SOTA systems.

### 4.3 Ensemble Analysis on WMT18 QE Task

As we discussed previously, we tried both word and BPE tokenization for the data pre-processing. Thus, we submitted two types of ensemble models, where Ensemble 1 is referred to the model ensembles trained with word tokenization and Ensemble 2 is the model ensembles trained with both word and BPE tokenizations. Training with BPE tokenization can naturally increase the model diversity, so it makes sense that Ensemble 2 performs better than Ensemble 1, except for English-German NMT word-level task, which is very likely due to the small data size ($<$14000).

## 5 Conclusion

This paper introduces our machine translation quality estimation system, QE Brain, for both the sentence-level and word-level tasks in WMT 2018 Quality Estimation. The system proposes the neural Bilingual Expert model to extract semantic features from both the source and translation output for estimating translation quality with a bidirectional LSTM predictive model. In particular, three important strategies are utilized for obtaining positive results as incorporating human-crafted features, artificial QE data augmentation for more diversified training data and model ensemble with a greedy algorithm. The results of our system obtained No.1. in the English-German SMT scoring and ranking tasks as well as the German-English SMT ranking tasks. Furthermore, our system also produced the best results in all word-level English-German and German-English word and gap prediction tasks.

## References

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt

Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 169–214.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Zhiming Chen, Yiming Tan, Chenlin Zhang, Qingyu Xiang, Lilin Zhang, Maoxi Li, and Mingwen Wang. 2017. Improving machine translation quality estimation with neural network features. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 551–555.

Kai Fan, Bo Li, Fengming Zhou, and Jiayi Wang. 2018. "bilingual expert" can find translation errors. *arXiv preprint arXiv:1807.09433*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. *CoRR*, abs/1605.04800.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 562–568.

Anna Kozlova, Mariya Shmatova, and Anton Frolov. 2016. YSDA participation in the wmt'16 quality estimation shared task. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 793–799.

André FT Martins, Fabio Kepler, and Jose Monteiro. 2017. Unbabel's participation in the wmt17 translation quality estimation shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 569–574.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318.

Ioannis Partalas, Grigorios Tsoumakas, and Ioannis P. Vlahavas. 2008. Focused ensemble selection: A diversity-based method for greedy ensemble selection. In *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*, pages 117–121.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, System Demonstrations*, pages 115–120.

# Quality Estimation with Force-Decoded Attention and Cross-lingual Embeddings

**Elizaveta Yankovskaya** **Andre Tättar** **Mark Fishel**
Institute of Computer Science
University of Tartu, Estonia
{elizaveta.yankovskaya,andre.tattar,fishel}@ut.ee

## Abstract

This paper describes the submissions of the team from the University of Tartu for the sentence-level Quality Estimation shared task of WMT18. The proposed models use features based on attention weights of a neural machine translation system and cross-lingual phrase embeddings as input features of a regression model. Two of the proposed models require only a neural machine translation system with an attention mechanism with no additional resources. Results show that combining neural networks and baseline features leads to significant improvements over the baseline features alone.

## 1 Introduction

Over the last several years the quality of machine translation has grown significantly. However even today most machine translation systems produce a lot of unreliable translations, with translation quality varying greatly between different input and output segments. To estimate the quality of these translations several methods have been proposed (Specia et al., 2013; Martins et al., 2017; Kim et al., 2017a,b).

In this article we propose an approach to quality estimation that is based on a regression model with different sets of features stemming from the internal parameters of a neural machine translation (NMT) system. We investigate how different input features of the regression model affect the correlation between the automatic quality estimation score and human assessment. We show that our models work for any translation output, without access to the translation system that produced the translations in question.

## 2 Method

The main idea of our method is to use features based on NMT attention weights and metrics based on cross-lingual embeddings as features of a regression model. In the following we explain the details of both these feature sources.

### 2.1 Attention Weights

The encoder-decoder NMT systems with an attention mechanism (Bahdanau et al., 2014) produce the translation output with the help of computed attention weights showing the strength of the connection between the input and output tokens. These attention weights resemble a soft alignment and their visualization often clearly indicates the translation quality that can be expected – see Figure 1 for an example of a well translated sentence.

Rikters and Fishel (2017) have shown that the attention weights can be used for confidence estimation, but only if these attention weights were computed along with translations, using the internal parameters of the NMT system producing the translations. We expand their approach to apply attention weights to any translations, regardless of whether they were produced by a data-driven, rule-based translation system or even a human translator. The same approach is used for quality estimation in (Yankovskaya and Fishel, 2018).

To get attention weights for any translation pair, we replace the decoding part of the NMT system with computing the probability of the given translation under an NMT model for that language. This way beam search and selecting the output token with the highest predicted probability is replaced with selecting the next given output token; in other words, force-decoding is done. Thus, we can get attention weights for any source/translation pair without even knowing anything about the system that produced this translation output.

To get features for a regression model we have computed the following metrics proposed by Rik-

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 816–821
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64093

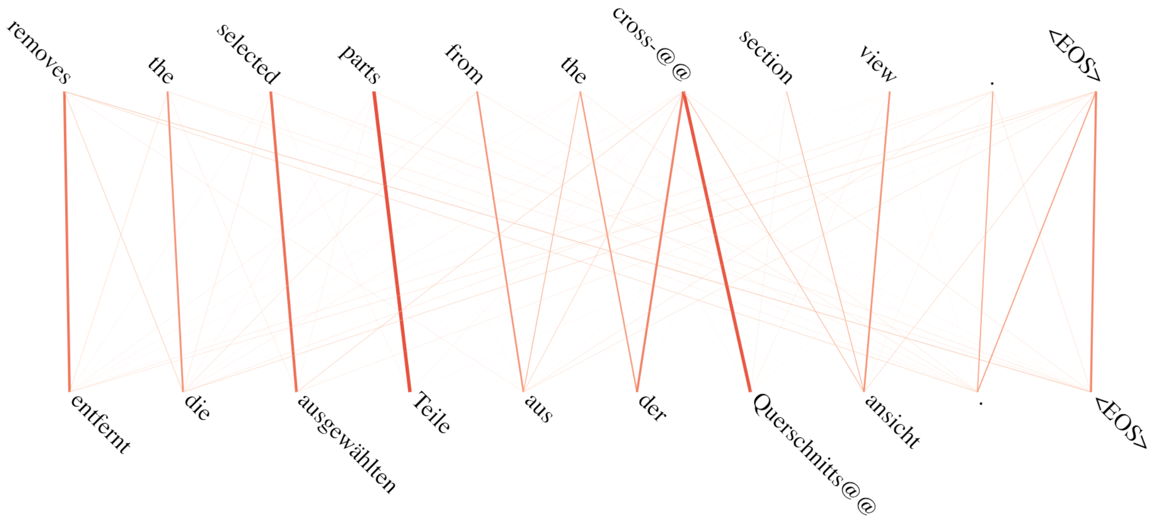


Figure 1: Attention alignment visualization of a well translated sentence from English to German. The thicker the line, the stronger the connection between the tokens (Rikters et al., 2017). It is visible from the alignment visualization alone that the quality/confidence of the translation system is high: each input/output token has a strong connection to one or at most two tokens on the other side.

ters and Fishel (2017) (see their paper for a more detailed definition):

- **Coverage Deviation Penalty (CDP)** penalizes the sum of attentions per input token, so tokens with less or too much attention get lower scores.
- **Absentmindedness Penalties (APin and APout)** compute the dispersion via the entropy of the attention distribution of input and output tokens.
- **Total** is the sum of all three metrics described above.

In addition to the metrics above we have calculated the ratio between input and output absentmindedness penalties as a small modification.

### 2.2 Cross-lingual Embeddings

NMT attention weights show the strength of the connection between the input and output tokens, but require running each segment pair through the NMT system. Here we try to align the input and output embeddings directly with the same aim of estimating the similarity between the input and output segments. This is done by taking the embedding-enhanced BLEU score called BLEU2VEC (Tättar and Fishel, 2017) and doing it cross-lingually.

We used three different types of embeddings to learn the cross-lingual similarity:

- **Word**-level embeddings were trained on tokenized data that consisted only of unigram words.
- **Phrase**-level embeddings were trained on data that concatenated words into phrases stochastically (Tättar and Fishel, 2017). Phrases consisted of up to three words concatenated with underscores.
- **BPE**-level embeddings use the embeddings from NMT systems that are trained on byte pair encoded data (Sennrich et al., 2015). BPE (byte-pair encoding) splits words into sub-word units in order to reduce the number of unique tokens.

The word-level and phrase-level embeddings were trained separately using monolingual corpora.Embeddings for BPE-s came from the attention-decoder translation system used in the attention weight feature extraction. These embeddings were not trained separately, so no additional training time was required for them.

After learning the monolingual embeddings, joint cross-lingual vector spaces are learned based on the monolingual ones, using the method of (Conneau et al., 2017). Cross-lingual mappings are learned between all the language pairs using MUSE[1]. In case of word-level and phrase-level mappings we used the supervised learning which

[1] A library for Multilingual Unsupervised or Su-

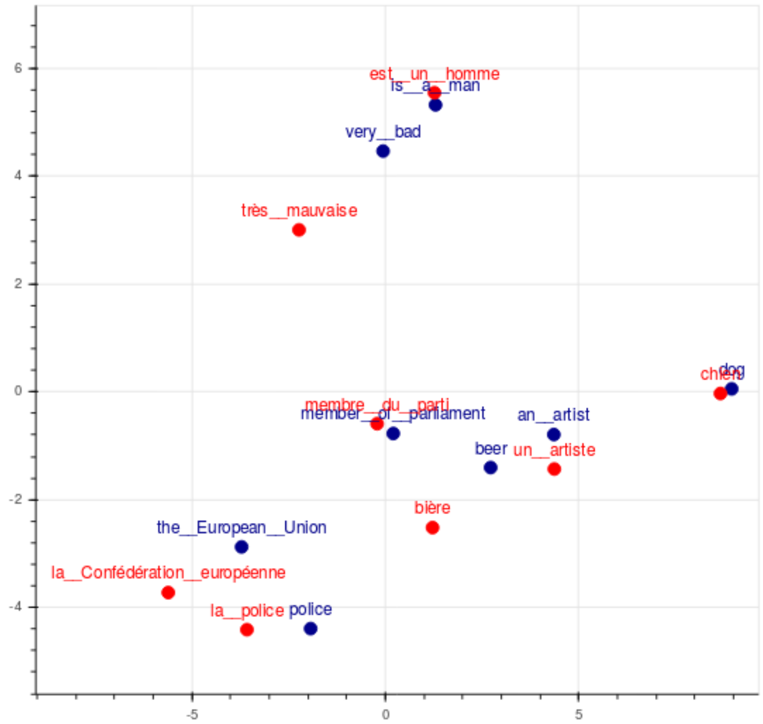


Figure 2: An examples of cross-lingual embeddings for French and English in the same vector space. On the figure, the closest neighbor was found and put on the graph. Dimensions are reduced from 300 down to the 2 first PCA components. Phrases are concatenated with two underscores. Blue means the source word/phrase and red means the nearest neighbor.

uses a seed dictionary of 1500 words for learning the mapping. For BPE embeddings we used the unsupervised cross-lingual mapping, which does not require a seed dictionary. Both methods of learning cross-lingual mappings for embedding spaces are described in (Conneau et al., 2017).

With the cross-lingual embeddings ready we compute the BLEU2VEC score:

- we find the optimal alignment between the words, n-grams or subwords of the input and output segments using beam search
- using this alignment we compute the BLEU score's (Papineni et al., 2002) n-gram precisions, giving partial credit to aligned n-gram (or word/sub-word) pairs equal to the cosine similarity of their cross-lingual embeddings

We can see examples of words/phrases after training cross-lingual embeddings in Figure 2. The nearest neighbor for a source word or phrase is visualized in the figure, which can be words or phrases in target language.

## 3 Experimental Settings

### 3.1 Data

We have applied our methods to all language pairs presented in the WMT18 shared task on sentence-level quality estimation (Specia et al., 2018): German-English, English-German, English-Latvian and English-Czech. For English-German and English-Latvian language pairs the translation output was produced by NMT and SMT systems, for other languages only SMT translations were given.

The number of sentences for each language pair and each machine translation system is shown in Table 1.

### 3.2 Experiments

The main goal of our experiments is to predict the normalized edit distance (HTER) (Snover et al., 2006). To estimate the quality of prediction we used the Pearson correlation coefficient.

As a regression model we used Random Forest (Ho, 1995) with a grid search algorithm for the optimization of parameters.

To get force-decoded attention weights and

pervised word Embeddings, https://github.com/facebookresearch/MUSE

| | EN-DE | | DE-EN | | EN-CS | | EN-LV | |
|---|---|---|---|---|---|---|---|---|
| | nmt | smt | nmt | smt | nmt | smt | nmt | smt |
| train | 13442 | 26299 | - | 26032 | - | 40254 | 12936 | 11251 |
| dev | 1000 | 1000 | - | 1000 | - | 1000 | 1000 | 1000 |
| test | 1023 | 1926 | - | 1254 | - | 1920 | 1448 | 1315 |

Table 1: Number of sentences for each language pair and each machine translation system.

BPE embeddings for all language pairs we used NMT models trained by the University of Edinburgh (Sennrich et al., 2017) for English-German, German-English and English-Czech; for English-Latvian we used a different NMT model trained separately.

Our chosen implementation of word and phrase embeddings was FastText (Bojanowski et al., 2016) with a continuous bag-of-words (CBOW) model and the number of dimensions for embeddings was set to 300. MUSE (Conneau et al., 2017) was used for extracting cross-lingual embeddings, with default parameters. A simple beam search was implemented for finding the quality estimation BLEU2VEC score, with beam size 3.

Initial tests showed that models with features based on cross-lingual embeddings only gave a close-to-zero Pearson correlation score, therefore these were not included as standalone features into the final experiments. A combination of cross-lingual embeddings (words, phrases, BPE) demonstrated a little bit better results but they were still lower than results obtained by using a model based on the attention weights. Taking into the account these results, we ran the final experiments with the following sets of features:

- **QuEst**: a standard set of 17 black-box QuEst features (Specia et al., 2013);
- **AttW**: features based on the force-decoded attention weights: $CDP$, $AP_{in}$, $AP_{out}$, $total$, $AP_{ratio}$;
- **QuEst+AttW**: a combination of QuEst and attention weights features;
- **QuEst+AttW+CrEmb3**: a combination of QuEst, attention weights and cross-lingual embeddings (phrases, words and BPE) features;
- **AttW+BPE**: a combination of attention weights and cross-lingual embeddings (BPE) features – to test a scenario of using only the parameters of an NMT system, both for the attention weights and the BPE embeddings
- **AttW+CrEmb3**: a combination of attention weights and cross-lingual embeddings (phrases, words and BPE) features.

The model with QuEst features was used as a baseline.

## 4 Results

The resulting Pearson coefficients for the dev and test sets for the all given language pairs are presented in Table 2. As one can see the highest values were obtained by applying the models `QuEst+AttW` or `QuEst+AttW+CrEmb3`. For English-German (NMT and SMT) and English-Latvian (SMT) language pairs the difference between these two models is negligible.

The baseline model shows the best result for all language pairs but German-English in comparison with two of our models: `AttW` and `AttW+BPE`. Although for English-Czech and English-Latvian (NMT) the difference between the baseline model and our models is small: 0.389/0.355 and 0.462/0.445. It is interesting to note that for German-English all of our proposed models showed a result that is more than twice the baseline model's result.

The main advantage of our models `AttW` and `AttW+BPE` is that they do not require additional resources like language models, n-gram frequencies, alignment probability files or even additional embedding models. In the case when the translation output is produced by an NMT system with an attention mechanism both models require attention weights or/and BPE embeddings of this NMT model. In the case when the system produced the translation is unknown one might use any NMT system with an attention mechanism.

|  | EN-DE |  |  |  | DE-EN |  | EN-CS |  | EN-LV |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | smt |  | nmt |  | smt |  | smt |  | smt |  | nmt |  |
|  | dev | test | dev | test | dev | test | dev | test | dev | test | dev | test |
| QuEst | 0.387 | 0.369 | 0.390 | 0.354 | 0.392 | 0.220 | 0.406 | 0.389 | 0.382 | 0.389 | 0.491 | 0.462 |
| AttW | 0.292 | 0.249 | 0.197 | 0.219 | 0.539 | 0.533 | 0.313 | 0.319 | 0.336 | 0.323 | 0.394 | 0.438 |
| AttW+ BPE | 0.303 | 0.260 | 0.207 | 0.230 | 0.553 | 0.544 | 0.326 | 0.355 | 0.357 | 0.323 | 0.403 | 0.445 |
| AttW+ CrEmb3 | 0.303 | 0.209 | 0.244 | 0.224 | 0.559 | 0.551 | 0.353 | 0.250 | 0.349 | 0.323 | 0.454 | 0.444 |
| QuEst+ AttW | 0.453 | **0.426** | 0.405 | **0.373** | 0.565 | 0.554 | 0.468 | **0.451** | 0.460 | 0.402 | 0.562 | 0.531 |
| QuEst+ AttW+ CrEmb3 | 0.457 | 0.424 | 0.408 | 0.369 | 0.592 | **0.570** | 0.487 | 0.406 | 0.461 | **0.404** | 0.585 | **0.542** |

Table 2: The Pearson correlation coefficients for the dev and test sets for all language pairs.

## 5 Discussions

As we mentioned above, the value of the Pearson correlation coefficient for German-English language pair is much higher than the values for other language pairs. A similar result is observed for the data of the last year Quality Estimation shared task, where the resulting Pearson correlation coefficient produced by the model `AttW` was 0.302 for English-German and 0.485 for German-English. We assume that this is related to the domain of data: German-English and English-Latvian data belongs to one domain (pharmaceutical) whereas English-German and English-Czech sentences were taken from the another domain (IT). This assumption is confirmed by the fact that the values of the Pearson correlation coefficient for English-Latvian are also slightly higher than the values for other language pairs.

To investigate how the choice of the NMT system affects the Pearson correlation between an automatic prediction and human assessment, we compared the results of our NMT system and University of Edinburgh's NMT system for German-English language pair.

The resulting Pearson coefficients of two proposed models `AttW` and `QuEst+AttW` are presented in Table 3. The resulting scores differ but not significantly; although on one hand this suggests that the choice of the NMT system is not important, both of the compared NMT systems are general-domain models, equally dissimilar from both of the test data domains; a more thorough comparison is left for future explorations.

|  | AttW |  | QuEst +AttW |  |
|---|---|---|---|---|
|  | dev | test | dev | test |
| Edinburgh's NMT system | 0.539 | 0.533 | 0.565 | 0.554 |
| Our NMT system | 0.560 | 0.562 | 0.594 | 0.584 |

Table 3: The Pearson coefficients of two regression models for German-English language pair. Attention weights were obtained from two different systems.

## 6 Conclusions

In this paper we described our submissions to the sentence-level subtask of WMT18 Quality Estimation task. We proposed several models for quality estimation of machine translation based on attention weights and embeddings. Our models do not require any additional resources, except for an NMT system and/or cross-lingual word embeddings learned from monolingual corpora. In the case when the translation output is produced by an NMT system with an attention mechanism, two of our models require only attention weights and BPE embeddings that are already created by this system.

For several language pairs the proposed models demonstrated comparable results with the baseline model. In the case of the German-English language pair all of our systems showed a much better result compared to the baseline model. Furthermore, the combination of neural networks and baseline features gave much better results than the results of the baseline model.

We plan to further experiment with the attention weights for in-domain systems and compare the scores obtained by using the internal and force-decoded attention weights.

## Acknowledgments

This work was supported by the Estonian Research Council grant no. 1226.

The authors would like to thank Matīss Rikters for providing the English-Latvian NMT system used in the experiments.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *CoRR*, abs/1710.04087.

Tin Kam Ho. 1995. Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, volume 1, pages 278–282. IEEE.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017a. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(1):3.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017b. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568.

André F. T. Martins, Fabio Kepler, and Jose Monteiro. 2017. Unbabel's participation in the wmt17 translation quality estimation shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 569–574, Copenhagen, Denmark.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Matīss Rikters and Mark Fishel. 2017. Confidence through attention. In *Proceedings of MT Summit XVI*, pages 299–311, Nagoya, Japan.

Matīss Rikters, Mark Fishel, and Ondřej Bojar. 2017. Visualizing Neural Machine Translation Attention and Confidence. volume 109, pages 1–12, Lisbon, Portugal.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's Neural MT Systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, Copenhagen, Denmark.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón F. Astudillo, and André Martins. 2018. Findings of the wmt 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Lucia Specia, Kashif Shah, Jose GC Souza, and Trevor Cohn. 2013. Quest-a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84.

Andre Tättar and Mark Fishel. 2017. bleu2vec: the painfully familiar metric on continuous vector space steroids. In *Proceedings of the Second Conference on Machine Translation*, pages 619–622, Copenhagen, Denmark. Association for Computational Linguistics.

Elizaveta Yankovskaya and Mark Fishel. 2018. Low-resource translation quality estimation for estonian. In *Proceedings of BalticHLT: the 8th International Conference Human Language Technologies: the Baltic Perspective*, Tartu, Estonia.

# MS-UEdin Submission to the WMT2018 APE Shared Task: Dual-Source Transformer for Automatic Post-Editing

**Marcin Junczys-Dowmunt**
Microsoft
Redmond, WA 98052, USA
marcinjd@microsoft.com

**Roman Grundkiewicz**
University of Edinburgh
10 Crichton St, Edinburgh EH8 9AB, Scotland
rgrundki@inf.ed.ac.uk

## Abstract

This paper describes the Microsoft and University of Edinburgh submission to the Automatic Post-editing shared task at WMT2018. Based on training data and systems from the WMT2017 shared task, we re-implement our own models from the last shared task and introduce improvements based on extensive parameter sharing. Next we experiment with our implementation of dual-source transformer models and data selection for the IT domain. Our submissions decisively wins the SMT post-editing sub-task establishing the new state-of-the-art and is a very close second (or equal, 16.46 vs 16.50 TER) in the NMT sub-task. Based on the rather weak results in the NMT sub-task, we hypothesize that neural-on-neural APE might not be actually useful.

## 1 Introduction

This paper describes the Microsoft (MS) and University of Edinburgh (UEdin) submission to the Automatic Post-editing shared task at WMT2018 (Chatterjee et al., 2018). Based on training data and systems from the WMT2017 shared task (Bojar et al., 2017), we re-implement our own models from the last shared task (Junczys-Dowmunt and Grundkiewicz, 2017a,b) and introduce a few small improvements based on extensive parameter sharing. Next, we experiment with our implementation of dual-source transformer models which have been available in our NMT toolkit Marian (Junczys-Dowmunt et al., 2018) since version v1.0 (November 2017). We believe this is one of the first descriptions of such an architectures for Automatic Post-Editing (APE) purposes, but similar approaches have been used for two-step decoding, for instance in Hassan et al. (2018). We further extend this model to share parameters across encoders with improved results for APE.

Our submissions decisively wins the SMT post-editing sub-task establishing the new state-of-the-art and is a very close second (or equal, 16.46 vs 16.50 TER) in the NMT sub-task.[1]

## 2 Training, development, and test data

We perform all our experiments with the official WMT-2018 automatic post-editing data and the respective development and test sets. The training data consists of a small set of post-editing triplets $(src, mt, pe)$, where $src$ is the original English text, $mt$ is the raw MT output generated by an English-to-German system, and $pe$ is the human post-edited MT output. The MT system used to produce the raw MT output is unknown, as is the original training data. The task consists of automatically correcting the MT output so that it resembles human post-edited data. The main task metric is TER (Snover et al., 2006) — the lower the better — with BLEU (Papineni et al., 2002) as a secondary metric.

To overcome the problem of too little training data, Junczys-Dowmunt and Grundkiewicz (2016) — the authors of the best WMT-2016 APE shared task system — generated large amounts of artificial data via round-trip translations. The artificial data has been filtered to match the HTER statistics of the training and development data for the shared task and was made available for download.

The organizers also made available a large new resource for APE training, the eSCAPE corpus (Negri et al., 2018), which contains triplets generated from SMT and NMT systems in separate data sets.

To produce our final training data set we oversample the original training data 20 times and add both artificial data sets. This results in a total of

[1] We did not make the models available, but researchers interested in reproducing these results are encouraged to contact one or both of the authors. We will be happy to help. The used architectures are available in Marian: https://marian-nmt.github.io

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 822–826
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64094

slightly more than 5M training triplets. We validate on the development set for early stopping and report results on the WMT-2016 APE test set. The data is already tokenized. Additionally we truecase all files and apply segmentation into BPE subword units (Sennrich et al., 2016). We reuse the subword units distributed with the artificial data set.

## 3 Experiments

During the WMT2017 APE shared task we submitted a dual-source model with soft and hard attention which placed second right after a very similar dual-source model by the FBK team. We include the performance of those models based on the shared task descriptions in Table 1, systems WMT17:FBK and WMT17:AMU (ours).

We mostly worked on the APE sub-task for automatic post-editing for the SMT system. The system in the NMT sub-task seemed to have only small margins for improvements.

### 3.1 Baselines

During the WMT2017 shared task on post-editing we made an error in judgment and submitted the weaker hard-attention model, in post-submission experiments we saw that a normal soft-attention model would have fared better. This was confirmed by the shared-task winner FBK and our own experiments. For this year, we first recreated our own dual-source model with soft attention (Baseline) and further experimented with parameter sharing:

- We first tie embeddings across all encoder instances, the decoder embedding layer and decoder output layer (transposed). This leads to visible improvements over our baseline across all test sets in terms of TER.
- Next, we share all parameters across encoders, despite the fact that these are encoding different language it seems that parameter sharing is generally beneficial. We see improvement across two test sets and roughly equal performance for the third.

### 3.2 Dual-source transformer

Figure 1 illustrates the architecture of our dual-source transformer variant. We naturally extend the original architecture from Vaswani et al. (2017) by adding another encoder and stacking an additional target-source multi-head attention component above the previous target-source multi-head attention component. This results in one target-source attention component per block for each encoder. As usual for the transformer architecture, each multi-head attention block is followed by a skip connection from the previous input and layer normalization. Each encoder corresponds exactly to the implementation from Vaswani et al. (2017), but with common parameters. Apart from these modifications, we follow the transformer-base configuration from Vaswani et al. (2017). This means that we tie source, target and output embeddings.

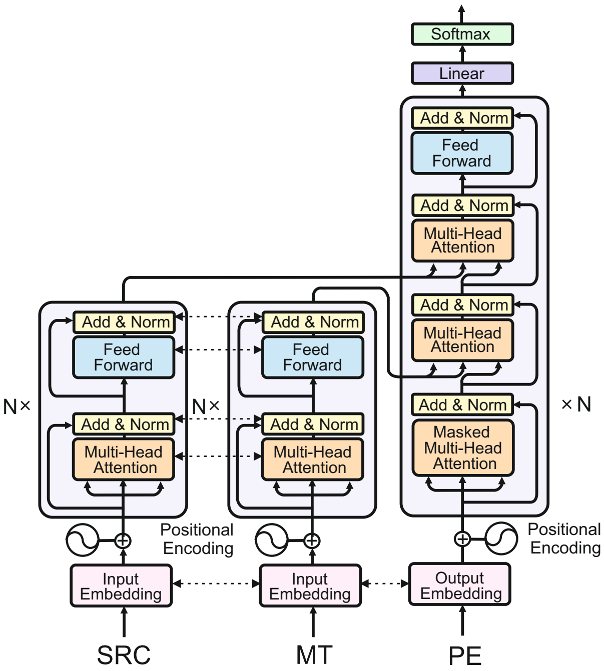


Figure 1: Dual-source transformer architecture. Dashed arrows mark tied parameters between the two separate encoders and common embedding matrices for all encoders and the decoder.

We found earlier that sharing parameters between the encoders is beneficial for the APE task and apply the same modification to our architecture, marked by dashed arrows in Figure 1. The two encoders share all parameters, but still produce different activations and are combined in different places in the decoder.

We briefly experimented with concatenating the encoder outputs instead of stacking (this would have been more similar to our work in Junczys-Dowmunt and Grundkiewicz (2017a,b)), but found this solution to underperform. We also replaced skip connections with gating mechanisms, but did not see any improvements.

The transformer architecture with its skip connections and normalization blocks can be seen to

|  | dev 2016 |  | test 2016 |  | test 2017 |  |
|---|---|---|---|---|---|---|
| Model | **TER**↓ | BLEU↑ | **TER**↓ | BLEU↑ | **TER**↓ | BLEU↑ |
| Uncorrected | 24.81 | 62.92 | 24.76 | 62.11 | 24.48 | 62.49 |
| WMT17: FBK Primary | 19.22 | 71.89 | 19.32 | 70.88 | 19.60 | 70.07 |
| WMT17: AMU Primary | — | — | 19.21 | 70.51 | 19.77 | 69.50 |
| Baseline (single model) | 19.77 | 70.54 | 20.10 | 69.25 | 20.43 | 68.48 |
| +Tied embeddings | 19.39 | 70.70 | 19.82 | 68.87 | 20.09 | 69.06 |
| +Shared encoder | 19.23 | 71.14 | 19.44 | 70.06 | 20.15 | 69.04 |
| Transformer-base (Tied+Shared) | 18.73 | 71.71 | 18.92 | 70.86 | 19.49 | 69.72 |
| Transformer-base x4 | 18.22 | 72.34 | 18.86 | 71.04 | 19.03 | 70.46 |

Table 1: Experiments with WMT 2017 data, correcting a phrase-base system.

|  | dev 2016 |  | test 2016 |  | test 2017 |  |
|---|---|---|---|---|---|---|
| Model | **TER**↓ | BLEU↑ | **TER**↓ | BLEU↑ | **TER**↓ | BLEU↑ |
| Transformer all | 17.84 | 73.45 | 17.81 | 72.79 | 18.10 | 71.72 |
| Transformer 1M | 17.59 | 73.45 | 18.29 | 72.20 | 18.42 | 71.50 |
| Transformer 2M | 17.92 | 73.37 | 18.02 | 72.41 | 18.35 | 71.57 |
| Transformer 4M | 17.75 | 73.51 | 17.89 | 72.70 | 18.09 | 71.78 |
| **Transformer x4 (all above)** | **17.31** | **74.14** | **17.34** | **73.43** | **17.47** | **72.84** |

Table 2: Experiments with WMT 2017+eSCAPE data for SMT system.

learn interpolation functions between layers that are not much different from gating mechanisms.

A single model of this type outperforms already the complex APE ensembles from the previous shared task in terms of TER and is on par in terms of BLEU (Table 1). An ensemble of four identical models trained with different random initializations strongly improves over last year's best models on all indicators.

### 3.3 Experiments with eSCAPE

So far, we only trained on data that was available during WMT2017. This year, the task organizers added a new large corpus created for automatic post-editing across many domains. We experimented with domain selection algorithms for this corpus and tried to find subsets that would be better suited to the given IT domain. We trained an 5-gram language model on a 10M words randomly sampled subset of the German IT training data and a similarly size language model on the eSCAPE data. Next we applied cross-entropy filtering (Moore and Lewis, 2010) to produce domain scores. We sorted eSCAPE by these scores and selected different sizes of subsets. Smaller subsets should be more in-domain. We experimented with 1M, 2M, 4M and all sentences (nearly 8M). Results (Table 2) remain however inconclusive. Adding eSCAPE to the training data was generally helpful, but we did not see a clear winner across subsets and test sets. In the end we use all the experimental models as components of a 4x ensemble. The different training sets might as well serve as additional randomization factors potentially beneficial for ensembling.

### 3.4 The NMT APE sub-task

So far we reported only results for the SMT APE sub-task. For the NMT system we trained our transformer-base model on eSCAPE NMT data only. Including SMT-specific data seemed to be harmful. In the end we only applied an ensemble of 4 such models observing moderate improvements on the development data. It seemed that our system was quite good at correcting errors due to hallucinated BPE words. We believe that our shared embeddings/encoders were helpful here. This does however indicate that the corrected NMT system was not well designed as these errors could have been easily avoided by the original MT system.

| Systems | TER↓ | BLEU↑ |
|---|---|---|
| **MS-UEdin (Ours)** | **18.00** | **72.52** |
| FBK | 18.62 | 71.04 |
| POSTECH | 19.63 | 69.87 |
| USAAR DFKI | 22.69 | 66.16 |
| DFKI-MLT | 24.19 | 63.40 |
| Baseline | 24.24 | 62.99 |

(a) PBSMT sub-task

| Systems | TER↓ | BLEU↑ |
|---|---|---|
| FBK | 16.46 | 75.53 |
| **MS-UEdin (Ours)** | **16.50** | **75.44** |
| POSTECH | 16.70 | 75.14 |
| Baseline | 16.84 | 74.73 |
| USAAR DFKI | 17.23 | 74.22 |
| DFKI-MLT | 18.84 | 70.87 |

(b) NMT sub-task

Table 3: APE Results provided by shared task organizers. We only include best-scored results by each team, see Chatterjee et al. (2018) for the full list of results.

Furthermore, our submission did only train for about one day, we would expect better results for a converged system, but we did not pursue this any further due to time constraints.

## 4 Results and conclusions

The organizers informed us about the results of our systems and we include the scores for the best system of each team in Table 3. For full results with information concerning statistical significance see the full shared task description (Chatterjee et al., 2018). As expected, improvements are quite significant for the SMT-based system, and much smaller for the NMT-based system. Our submissions to the PBSMT sub-task strongly outperforms all submissions by other teams in terms of TER and BLEU and established the new state-of-the-art for the field. The improvements over the PBSMT baseline approach impressive 10 BLEU points.

For the NMT sub-task our submission places second with a 0.04 TER difference behind the leading submission. We would call this an equal result. This is interesting considering how little time and effort was spent on our NMT system compared to the SMT system. One day more or training time might have flipped these results.

Based on the overall weak performance for the neural sub-task, we feel justified in not investing much time into that particular sub-task. We hypothesize that if the same amount of effort had been put into the NMT baseline as into the APE systems that were submitted to the task, none of the submissions (including our own) would have been able to beat that baseline. We saw obvious problems with BPE handling in the baseline which could have been easily fixed. It is probable that most of our improvements come from correcting those BPE errors.

We further believe that this might constitute the end of neural automatic post-editing for strong neural in-domain systems. The next shared task should concentrate on correcting general domain on-line systems. Another interesting path would be to make the original NMT training data available so that both, pure NMT systems and APE systems, can compete. This would show us where we actually stand in terms of feasibility of neural-on-neural automatic post-editing.

## Acknowledgments

We would like to thank Elena Voita and Rico Sennrich for sharing their transformer figures which we used as basis for our dual-source transformer illustration. We also thank Kenneth Heafield for his comments on the paper.

This work was partially funded by Facebook. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Facebook.

## References

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers, pages 169–214, Copenhagen. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 Shared Task on Automatic Post-Editing. In Proceedings of the Third Conference on Machine

Translation, Volume 2: Shared Task Papers, Brussels, Belgium. Association for Computational Linguistics.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. CoRR, abs/1803.05567.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In Proceedings of the First Conference on Machine Translation, pages 751–758.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017a. The AMU-UEdin submission to the WMT 2017 shared task on automatic post-editing. In Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers, pages 639–646, Copenhagen. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017b. An exploration of neural sequence-to-sequence architectures for automatic post-editing. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 120–129. Asian Federation of Natural Language Processing.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In Proceedings of ACL 2018, System Demonstrations, pages 116–121. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In Proceedings of the ACL 2010 Conference Short Papers, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. eSCAPE: a large-scale synthetic corpus for automatic post-editing. CoRR, abs/1803.07274.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In Proceedings of Association for Machine Translation in the Americas,.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc.

# A Transformer-Based Multi-Source Automatic Post-Editing System

**Santanu Pal[1,2], Nico Herbig[2], Antonio Krüger[2], Josef van Genabith[1,2]**
[1]Department of Language Science and Technology,
Saarland University, Germany
[2]German Research Center for Artificial Intelligence (DFKI),
Saarland Informatics Campus, Germany
{santanu.pal, josef.vangenabith}@uni-saarland.de
{nico.herbig, krueger, josef.van_genabith}@dfki.de

## Abstract

This paper presents our English–German Automatic Post-Editing (APE) system submitted to the APE Task organized at WMT 2018 (Chatterjee et al., 2018). The proposed model is an extension of the transformer architecture: two separate self-attention-based encoders encode the machine translation output ($mt$) and the source ($src$), followed by a joint encoder that attends over a combination of these two encoded sequences ($enc_{src}$ and $enc_{mt}$) for generating the post-edited sentence. We compare this multi-source architecture (i.e, $\{src, mt\} \rightarrow pe$) to a monolingual transformer (i.e., $mt \rightarrow pe$) model and an ensemble combining the multi-source $\{src, mt\} \rightarrow pe$ and single-source $mt \rightarrow pe$ models. For both the PBSMT and the NMT task, the ensemble yields the best results, followed by the multi-source model and last the single-source approach. Our best model, the ensemble, achieves a BLEU score of 66.16 and 74.22 for the PBSMT and NMT task, respectively.

## 1 Introduction & Related Work

The ultimate goal of machine translation (MT) is to provide fully automatic publishable quality translations. However, state-of-the-art MT systems often fail to deliver this; translations produced by MT systems contain different errors and require human interventions to post-edit the translations. Nevertheless, MT has become a standard in the translation industry as post-editing on MT output is often faster and cheaper than performing human translation from scratch.

APE is a method that aims to automatically correct errors made by MT systems before performing actual human-post-editing (PE) (Knight and Chander, 1994), thereby reducing the translators' workload and increasing productivity (Parra Escartín and Arcedillo, 2015b,a; Pal et al., 2016a). Various automatic and semi-automatic techniques have been developed to auto-correct repetitive errors (Roturier, 2009; TAUS/CNGL Report, 2010). The advantage of APE lies in its capability to adapt to any black-box (first-stage) MT engine; i.e., upon availability of human-corrected post-edited data, no incremental training or full retraining of the first-stage MT system is required to improve the overall translation quality. APE can therefore be viewed as a 2nd-stage MT system, translating predictable error patterns in MT output to their corresponding corrections. APE training data minimally involves MT output ($mt$) and the human-post-edited ($pe$) version of $mt$, but may additionally make use of the source ($src$). A more detailed motivation on APE can be found in Bojar et al. (2015, 2016, 2017).

Based on the training process, APE systems can be categorized as either single-source ($mt \rightarrow pe$) or multi-source ($\{src, mt\} \rightarrow pe$) approaches. In general, the field of APE covers a wide methodological range, including SMT-based approaches (Simard et al., 2007a,b; Lagarda et al., 2009; Rosa et al., 2012; Pal et al., 2016c; Chatterjee et al., 2017b), and neural APE (Pal et al., 2016b; Junczys-Dowmunt and Grundkiewicz, 2016; Pal et al., 2017) based on neural machine translation (NMT). Some of the state-of-the-art multi-source approaches, both statistical (Béchara et al., 2011; Chatterjee et al., 2015) and recently neural (Libovický et al., 2016; Chatterjee et al., 2017a; Junczys-Dowmunt and Grundkiewicz, 2016; Varis and Bojar, 2017), explore learning from $\{src, mt\} \rightarrow pe$ (multi-source, MS)

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 827–835
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64095

to take advantage of the dependencies of translation errors in $mt$ originating from $src$.

Exploiting source information in multi-source neural APE can be configured either by using a single encoder that encodes the concatenation of $src$ and $mt$ (Niehues et al., 2016) or by using two separate encoders for $src$ and $mt$ and passing the concatenation of both encoders' final states to the decoder (Libovický et al., 2016). A few approaches to multi-source neural APE have been proposed in the WMT-2017 APE shared task. Junczys-Dowmunt and Grundkiewicz (2017) explore different combinations of attention mechanisms including soft attention and hard monotonic attention on an end-to-end neural APE model that combines both $mt$ and $src$ in a single neural architecture. Chatterjee et al. (2017a) extend the two-encoder architecture of multi-source models presented in Libovický et al. (2016). In their extension each encoder concatenates both contexts having their own attention layer that is used to compute the weighted context of $src$ and $mt$. Finally, a linear transformation is applied on the concatenation of both weighted contexts. Varis and Bojar (2017) implement and compare two multi-source architectures: In the first setup, they use a single encoder with concatenation of $src$ and $mt$ sentences, and in the second setup, they use two character-level encoders for $mt$ and $src$, separately, along with a character-level decoder. The initial state of this decoder is a weighted combination of the final states of the two encoders.

Intuitively, such an integration of source-language information in APE should be useful in conveying the context information to improve the APE performance. To provide the awareness of errors in $mt$ originating from $src$, the transformer architecture (Vaswani et al., 2017), which is built solely upon attention mechanisms (Bahdanau et al., 2015), makes it possible to model dependencies without regard to their distance in the input or output sequences and also captures global dependencies between input and output (for our case $src$, $mt$, and $pe$). The transformer architecture replaces recurrence and convolutions by using positional encodings on both the input and output sequences. The encoder and decoder both use multi-head (facilitating parallel computations) self-attention to compute representations of their corresponding inputs, and also compute multi-head vanilla-attentions between encoder and decoder representations.

Our APE system extends this transformer-based NMT architecture (Vaswani et al., 2017) by using two encoders, a joint encoder, and a single decoder. Our model concatenates two separate self-attention-based encoders ($enc_{src}$ and $enc_{mt}$) and passes this sequence through another self-attended joint encoder ($enc_{src,mt}$) to ensure capturing dependencies between $src$ and $mt$. Finally, this joint encoder is fed to the decoder which follows a similar architecture as described in Vaswani et al. (2017). The entire model is optimized as a single end-to-end transformer network.

## 2 Transformer-Based Multi-Source APE

MT errors originating from the input source sentences suggest that APE systems should leverage information from both the $src$ and $mt$, instead of considering $mt$ in isolation. This can help the model to disambiguate corrections applied at every time step. Generating the $pe$ output from $mt$ is greatly facilitated by the availability of $src$. To achieve benefits from both ***single-source*** ($\mathbf{mt} \rightarrow \mathbf{pe}$) and ***multi-source*** ($\{\mathbf{src}, \mathbf{mt}\} \rightarrow \mathbf{pe}$) APEs, our primary submission in the WMT 2018 shared task is an ensemble of these two models.

Transformer-based models are currently providing state-of-the-art performance in MT; hence, we want to explore a similar architecture for this year's APE task. We extend the transformer architecture to investigate how efficient this approach is in a multi-source scenario. In a MT task, it was already shown that a transformer can learn long-range dependencies. Therefore, we explore if we can leverage information from $src$ and $mt$ via a joint encoder through self-attention (see Section 2.2) to provide dependencies between $src$–$mt$ that are then projected to the $pe$.

To investigate this, we implement and evaluate three different models: a single-source approach, a multi-source approach, and an ensemble of both, described in more detail below.

### 2.1 Single-Source Transformer for APE ($\mathbf{mt} \rightarrow \mathbf{pe}$)

Our single-source model (SS) is based on an encoder-decoder-based transformer architecture (Vaswani et al., 2017). Transformer models can replace sequence-aligned recurrence entirely and follow three types of multi-head attention: encoder-decoder attention (also known as vanilla

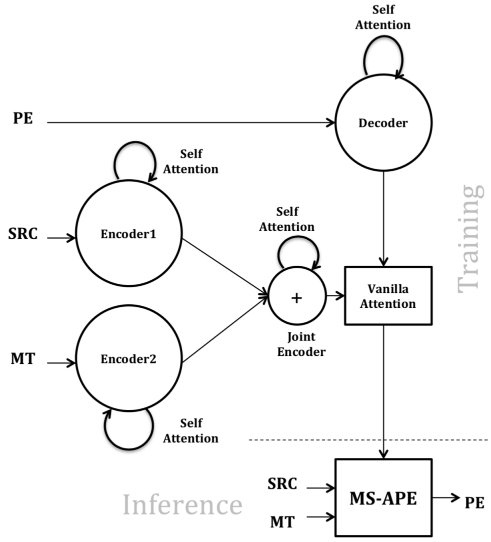


Figure 1: Multi-source transformer-based APE

attention), encoder self-attention, and masked decoder self-attention. Since for multi-head attention each head uses different linear transformations, it can learn these separate relationships in parallel, thereby improving learning time.

### 2.2 Multi-source Transformer for APE ($\{\mathbf{src}, \mathbf{mt}\} \rightarrow \mathbf{pe}$)

For our multi-source model (MS), we propose a novel joint transformer model (cf. Figure 1), which combines the encodings of $src$ and $mt$ and attends over a combination of both sequences while generating the post-edited sentence. Apart from $enc_{src}$ and $enc_{mt}$, each of which is equivalent to the original transformer's encoder (Vaswani et al., 2017), we use a joint encoder with an equivalent architecture, to maintain the homogeneity of the transformer model. For this, we extend Vaswani et al. (2017) by introducing an additional identical encoding block by which both the $enc_{src}$ and the $enc_{mt}$ encoders communicate with the decoder.

Our multi-source neural APE computes intermediate states $\mathbf{enc_{src}}$ and $\mathbf{enc_{mt}}$ for the two encoders, $\mathbf{enc_{src,mt}}$ for their combination, and $\mathbf{dec_{pe}}$ for the decoder in sequence-to-sequence modeling. One self-attended encoder for $src$ maps $\mathbf{s} = (s_1, s_2, ..., s_k)$ into a sequence of continuous representations, $\mathbf{enc_{src}} = (e_1, e_2, ..., e_k)$, and a second encoder for $mt$, $\mathbf{m} = (m_1, m_2, ..., m_l)$, returns another sequence of continuous representations, $\mathbf{enc_{mt}} = (e_1^{'}, e_2^{'}, ..., e_l^{'})$. The self-attended joint encoder (cf. Figure 1) then receives the concatenation of $\mathbf{enc_{src}}$ and $\mathbf{enc_{mt}}$, $\mathbf{enc_{concat}} = [\mathbf{enc_{src}}, \mathbf{enc_{mt}}]$ as an input, and passes it through the stack of 6 layers, with residual connections, a self-attention and a position-wise fully connected feed-forward neural network. As a result, the joint encoder produces a final representation ($\mathbf{enc_{src,mt}}$) for both $src$ and $mt$. Self-attention at this point provides the advantage of aggregating information from all of the words, including $src$ and $mt$, and successively generates a new representation per word informed by the entire $src$ and $mt$ context. The decoder generates the $pe$ output in sequence, $\mathbf{dec_{pe}} = (p_1, p_2, \ldots, p_n)$, one word at a time from left to right by attending previously generated words as well as the final representations ($\mathbf{enc_{src,mt}}$) generated by the encoder.

### 2.3 Ensemble

In order to leverage the network architecture for both single-source and multi-source APE as discussed above, we decided to **ensemble** several expert neural models. Each model is averaged using the 5 best saved checkpoints, which generate different translation outputs. Taking into account all these generated translation outputs, we implement an ensemble technique based on the frequency of occurrence of the output words. Corresponding to each input word, we calculate the most frequent occurrence of the output word from all the generated translation outputs. For the two different APE tasks, we ensemble the following models:

- PBSMT task: We ensemble a SS ($mt \rightarrow pe$) and a MS ($\{src, mt\} \rightarrow pe$) average model.
- NMT task: We ensemble two average SS ($mt \rightarrow pe$) and MS ($\{src, mt\} \rightarrow pe$) models, together with a SS and a MS model that are fine-tuned on a subset of the training set (cf. Section 3.3.2).

## 3 Experiments

In our experiment we investigate (1) how well the transformer-based APE architecture performs in general, (2) if our multi-source architecture using the additional joint encoder improves the performance over a single-source architecture, and (3) if ensembling of single-source and multi-source architectures facilitates APE even further.

### 3.1 Data

Since this year's WMT 2018 APE task (Chatterjee et al., 2018) is divided into two sub-tasks, different datasets are provided for each task: for the PBSMT task, there is a total of 23K English–German APE data samples (11K from WMT 2016 and 12K from WMT 2017) (Bojar et al., 2017). For the NMT task, 13,442 samples of English–German APE data are provided.

All released APE data consists of English–German triplets containing source English text ($src$) from the IT domain, the corresponding German translations ($mt$) from a first stage MT system, and the corresponding human-post-edited version ($pe$), all of them already tokenized. As this released APE dataset is small in size (see Table 1), additional resources are also available: first, the 'artificial training data' (Junczys-Dowmunt and Grundkiewicz, 2016) containing 4.5M sentences, 4M of which are weakly similar to the WMT 2016 training data, while 500K show very similar TER statistics; and second, the synthetic 'eSCAPE' APE corpus (Negri et al., 2018), consisting of more than 7M triples for both NMT and PBSMT.

Table 1 presents the statistics of the released data for the English–German APE Task organized in WMT 2018. These datasets, except for the eSCAPE corpus, do not require any preprocessing in terms of encoding or alignment.

For cleaning the noisy eSCAPE dataset containing many unrelated language words (e.g. Chinese), we perform the following two steps: (i) we use the cleaning process described in Pal et al. (2015), and (ii) we execute the Moses (Koehn et al., 2007) corpus cleaning scripts with minimum and maximum number of tokens set to 1 and 80, respectively. After cleaning, we use the Moses tokenizer to tokenize the eSCAPE corpus. To handle out-of-vocabulary words, words are preprocessed into subword units (Sennrich et al., 2016) using byte-pair encoding (BPE).

### 3.2 Hyper-Parameter Settings

For $\{\mathbf{src}, \mathbf{mt}\} \rightarrow \mathbf{pe}$, both the self-attended encoders, the joint encoder, and the decoder are composed of a stack of $N = 6$ identical layers followed by layer normalization. Each layer again consists of two sub-layers and a residual connection (He et al., 2016) around each of the two sub-layers. During training, we employ label smoothing of value $\epsilon_{ls}$ = 0.1. The output dimension produced by all sub-layers and embedding layers is defined as $d_{model} = 256$. All dropout values in the network are set to 0.2. Each encoder and decoder contains a fully connected feed-forward network having dimensionality $d_{model} = 256$ for the input and output and dimensionality $d_{ff} = 1024$ for the inner layer. This is a similar setting to Vaswani et al. (2017)'s $C - model$[1]. For the scaled dot-product attention, the input consists of queries and keys of dimension $d_k$, and values of dimension $d_v$. As multi-head attention parameters, we employ $h = 8$ for parallel attention layers, or heads. For each of these we use a dimensionality of $d_k = d_v = d_{model}/h = 32$. For optimization, we use the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. The learning rate is varied throughout the training process, first increasing linearly for the first training steps $warmup_{steps} = 4000$ and then adjusted as described in (Vaswani et al., 2017).

At training time, the batch size is set to 32 samples, with a maximum sentence length of 80 subwords, and a vocabulary of the 50K most frequent subwords. After each epoch, the training data is shuffled. For encoding the word order, our model uses learned positional embeddings (Gehring et al., 2017), since Vaswani et al. (2017) reported nearly identical results to sinusoidal encodings. After finishing training, we save the 5 best checkpoints saved at each epoch. Finally, we use a single model obtained by averaging the last 5 checkpoints. During decoding, we perform greedy-search-based decoding.

We follow a similar hyper-parameter setup for $mt \rightarrow pe$. The total number of parameters for our $\{src, mt\} \rightarrow pe$ and $mt \rightarrow pe$ model is $46 \times 10^6$ and $28 \times 10^6$, respectively.

### 3.3 Experiment Setup

In this section, we present the training process, using the above datasets, to train $mt \rightarrow pe$, $\{src, mt\} \rightarrow pe$, and ensemble models for both PBSMT and NMT.

#### 3.3.1 PBSMT Task

For PBSMT, we first train both our SS and MS systems with the cleaned eSCAPE corpus for 3 epochs. We then perform transfer learning with 4M artificial data for 7 epochs. Afterwards, fine-tuning is performed using the 500K artificial and 23K real PE training data for another 20 epochs.

[1] Note: at the time of submission we couldn't test the *Transformer (big)* model due to unavailability of enough computation power

|  | Sentences |  |  |  |  |
|---|---|---|---|---|---|
|  | Corpus | 2016 | 2017 | 2018 | Cleaning |
| PBSMT | Train | 12,000 | 11,000 | - | - |
|  | Dev | 1,000 | - | - | - |
|  | Test | 2,000 | 2,000 | 2,000 | - |
| NMT | Train | - | - | 13,442 | - |
|  | Dev | - | - | 1,000 | - |
|  | Test | - | - | 1,023 | - |
| Additional Resources | Artificial | - | 4M + 500K | - | - |
|  | eSCAPE-PBSMT | - | - | 7,258,533 | 6,521,736 |
|  | eSCAPE-NMT | - | - | 7,258,533 | 6,485,507 |

Table 1: Statistics of the WMT 2018 APE Shared Task Dataset.

Furthermore, we generate an ensemble model, by averaging the 5 best checkpoints of SS with the 5 best checkpoints of MS.

We use the WMT 2016 development data (dev2016) containing 1,000 triplets to validate the model during training. To test our system performance, we use the WMT 2016 and 2017 test data (test2016, test2017), each containing 2,000 triplets. Furthermore, we report the results of the submitted ensemble model on test2018.

### 3.3.2 NMT Task

Initial tests for pre-training our NMT model on the NMT eSCAPE data showed no performance improvements. Therefore, we use the PBSMT SS and MS models as a basis for the NMT task. We use the PBSMT models after training them on the eSCAPE corpus, the 4M artificial data and the 500K + 23K train sets of WMT 16 and 17. These SMT-based models are then fine-tuned using the WMT 2018 NMT APE data (train18) for 60 epochs.

Afterwards, we perform an additional fine-tuning step towards the dev18/test18 dataset: For this, we extract sentences of train18 that are similar to the sentences contained in dev18/test18 and fine-train for another 15 epochs on this subset of train18, which we call fine-tune18. As a similarity measure we use the cosine similarity between the train src and mt segments and the test src and mt segments, respectively. These cosine similarities for src and mt are then simply multiplied to achieve an overall similarity measure. Our fine-tuning dataset contains only sentences with an overall similarity of at least 0.9.

Last, two separate ensemble models are created. One consists of only the non-fine-tuned SS and MS models, and one ensembles the SS and MS models in both fine-tuned and non-fine-tuned variants. Both ensembles are created by averaging over the 5 best checkpoints of each sub-model.

We report the results of all created models for the dev18 NMT dataset, and additionally those of the submitted overall ensemble model on test18.

## 3.4 Results and Discussion

Table 2 presents the results for the PBSMT APE task (cf. 3.3.1), where two different transformer-based models, one ensemble of these models and the baseline BLEU scores are shown. The baseline here refers to the original MT output evaluated with respect to the corresponding PE translation. All models yield statistically significant results ($p < 0.001$) over this baseline. $MS_{avg}$ also provides statistically significant improvement over $SS_{avg}$. For this and all following significance tests we employ the method by Clark et al. (2011)[2].

Generally, reasons for the good performance of our transformer-based MS architecture in comparison to the SS approach for PBSMT-based APE could be the positional encoding that injects information about the relative or absolute position of the tokens in the sequence. This might help to handle word order errors in $mt$ for a given $src$ context. Another possible explanation lies in the self-attention mechanism, which handles local word dependencies for $src$, $mt$, and $pe$. After the individual dependencies are learned by the two encoders' self-attention mechanisms, another level of self-attention is performed that can jointly learn from both $src$ and $mt$ using our joint encoder, thereby informing the decoder about the long-range dependencies between the words within both $src$ and $mt$. Compared to RNNs, we believe that this technique can better convey source information via $mt$ to the decoder. The ensemble model then leverages the advantages of both our SS and MS approaches to further improve the results.

The results for our transformer-based architec-

[2] https://github.com/jhclark/multeval

| WMT APE Systems | eScape | 4M | 500K | train16 | train17 | test16 | test17 | test18 |
|---|---|---|---|---|---|---|---|---|
| Baseline | - | | | | | 62.92 | 62.11 | 62.99 |
| $MS_{avg}$ | 3 eps | 7 eps | 20 eps | | | 67.31 | 67.66 | - |
| $SS_{avg}$ | 3 eps | 7 eps | 20 eps | | | 66.27 | 66.60 | - |
| Ensemble | $MS_{avg\{5cps\}}$ + $SS_{avg\{5cps\}}$ | | | | | 68.52 | 68.91 | 66.16 |

Table 2: Evaluation result of WMT 2018 PBSMT task for all trained models.

| WMT APE Systems | Base Model | train18 | fine-tune18 | dev18 | test18 |
|---|---|---|---|---|---|
| Baseline | - | - | - | 76.66 | 74.73 |
| $MS_{avg}$ | $MS_{avg}$ (PBSMT) | 60 eps | - | 74.84 | - |
| $SS_{avg}$ | $SS_{avg}$ (PBSMT) | 60 eps | - | 72.75 | |
| $MS_{finetuned}$ | $MS_{avg}$ (NMT) | - | 15 eps | 75.05 | - |
| $SS_{finetuned}$ | $SS_{avg}$ (NMT) | - | 15 eps | 73.17 | - |
| $Ensemble$ | $MS_{avg\{5cps\}}$ + $SS_{avg\{5cps\}}$ | | | 75.80 | - |
| $Ensemble_{finetuned}$ | $MS_{avg\{5cps\}}$ + $SS_{avg\{5cps\}}$ + $MS_{finetuned\{5cps\}}$ + $SS_{finetuned\{5cps\}}$ | | | 75.96 | 74.22 |

Table 3: Evaluation result of WMT 2018 NMT task for all trained models.

ture for the NMT task are shown in Table 3. As can be seen, the baseline NMT system performs best, followed by the ensemble models, then the multi-source architectures and lastly the single-source approach. These differences between the three approaches, ensemble, MS, and SS, are all statistically significant. Fine-tuning provides some small, albeit insignificant, improvements over the non-fine-tuned versions.

While none of our architectures perform better than the baseline MT system for the NMT task, we clearly see that the multi-source approach helps, and that ensembling of different SS and MS models further increases the performance. These results are in line with our expectations, because intuitively, inspecting both $src$ and $mt$ should help detect and correct common errors. However, we are unsure why all models did not improve over the baseline, which could have been achieved by simply copying the $mt$. One reason might be the small amount of PE data, which comprises only 13K samples; this could also explain why the simple fine-tuning approach already leads to slightly higher BLEU scores. However, further human evaluation is necessary to better understand what our model is doing for the neural APE task and why it remains approximately 0.5 BLEU points below the baseline.

## 4 Conclusions and Future Work

In this paper, we investigated a novel transformer-based multi-source APE approach that jointly attends over a combination of $src$ and $mt$ to capture dependencies between the two. This architecture yields statistically significant improvements over single-source transformer-based models. An ensemble of both variants increases the performance further. For the PBSMT task, the baseline MT system was outperformed by 3.2 BLEU points, while the NMT baseline remains 0.51 BLEU points better than our APE approach on the 2018 test set.

In the future, we will investigate if the performance of each system can be improved by using a different hyper-parameter setup. Unfortunately, we could not test either the 'big' or the 'base' hyper-parameter configuration in Vaswani et al. (2017) due to unavailable computing resources at the time of submission. As additional future work, we would like to explore whether using re-ranking and ensembling of different neural APEs helps to improve the performance further. Moreover, we will incorporate word-level quality estimation features of $mt$ into the encoding layer. Lastly, we will evaluate if our model indeed is able to better handle word order errors and to capture long-range dependencies, as we expect. Furthermore, we will analyze if adapting the learning rate to the size of the datasets used during training increases the performance compared to the currently used fixed learning rate initialization of 0.001.

## Acknowledgments

This research was funded in part by the German research foundation (DFG) under grant number GE 2819/2-1 (project MMPE) and the German Federal Ministry of Education and Research (BMBF) under funding code 01IW17001 (project Deeplee). The responsibility for this publication lies with the authors. We also want to thank the reviewers for their valuable input, and the organizers of the shared task.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Hanna Béchara, Yanjun Ma, and Josef van Genabith. 2011. Statistical Post-Editing for a Statistical MT System. In *Proceedings of MT Summit XIII*, pages 308–315.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017a. Multi-source Neural Automatic Post-Editing: FBK's participation in the WMT 2017 APE shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 630–638, Copenhagen, Denmark. Association for Computational Linguistics.

Rajen Chatterjee, Gebremedhen Gebremelak, Matteo Negri, and Marco Turchi. 2017b. Online Automatic Post-editing for MT in a Multi-Domain Translation Environment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 525–535, Valencia, Spain. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 Shared Task on Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the Planet of the APEs: A Comparative Study of State-of-the-art Methods for MT Automatic Post-Editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156–161, Beijing, China.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. 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*, HLT '11, pages 176–181, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. *CoRR*, abs/1705.03122.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear Combinations of Monolingual and Bilingual Neural Machine Translation Models for Automatic Post-Editing. In *Proceedings of the First Conference on Machine Translation*, pages 751–758, Berlin, Germany.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. The AMU-UEdin Submission to the WMT 2017 Shared Task on Automatic Post-Editing. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 639–646, Copenhagen, Denmark. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. *ICLR*.

Kevin Knight and Ishwar Chander. 1994. Automated Postediting of Documents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 1)*, AAAI '94, pages 779–784, Seattle, Washington, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL: Interactive Poster and Demonstration Sessions*, pages 177–180, Prague, Czech Republic.

Antonio Lagarda, Vicent Alabau, Francisco Casacuberta, Roberto Silva, and Enrique Díaz-de Liaño. 2009. Statistical Post-editing of a Rule-based Machine Translation System. In *Proceedings of Human Language Technologies*, pages 217–220, Stroudsburg, PA, USA.

Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Ondřej Bojar, and Pavel Pecina. 2016. CUNI System for WMT16 Automatic Post-Editing and Multimodal Translation Tasks. In *Proceedings of the First Conference on Machine Translation*, pages 646–654, Berlin, Germany. Association for Computational Linguistics.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a Large-scale Synthetic Corpus for Automatic Post-Editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-Translation for Neural Machine Translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1828–1836, Osaka, Japan. The COLING 2016 Organizing Committee.

Santanu Pal, Sudip Naskar, and Josef van Genabith. 2015. UdS-Sant: English–German Hybrid Machine Translation System. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 152–157, Lisbon, Portugal. Association for Computational Linguistics.

Santanu Pal, Sudip Kumar Naskar, and Josef van Genabith. 2016a. Multi-Engine and Multi-Alignment Based Automatic Post-Editing and its Impact on Translation Productivity. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2559–2570, Osaka, Japan.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, and Josef van Genabith. 2016b. A Neural Network Based Approach to Automatic Post-Editing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 281–286, Berlin, Germany. Association for Computational Linguistics.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, Qun Liu, and Josef van Genabith. 2017. Neural Automatic Post-Editing Using Prior Alignment and Reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 349–355, Valencia, Spain. Association for Computational Linguistics.

Santanu Pal, Marcos Zampieri, and Josef van Genabith. 2016c. USAAR: An Operation Sequential Model for Automatic Statistical Post-Editing. In *Proceedings of the First Conference on Machine Translation*, pages 759–763, Berlin, Germany.

Carla Parra Escartín and Manuel Arcedillo. 2015a. Living on the Edge: Productivity Gain Thresholds in Machine Translation Evaluation Metrics. In *Proceedings of the Fourth Workshop on Post-editing Technology and Practice*, pages 46–56, Miami, Florida (USA). Association for Machine Translation in the Americas (AMTA).

Carla Parra Escartín and Manuel Arcedillo. 2015b. Machine Translation Evaluation Made Fuzzier: A Study on Post-Editing Productivity and Evaluation Metrics in Commercial Settings. In *Proceedings of the MT Summit XV*, Miami (Florida). International Association for Machine Translation (IAMT).

Rudolf Rosa, David Mareček, and Ondřej Dušek. 2012. DEPFIX: A System for Automatic Correction of Czech MT Outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 362–368, Stroudsburg, PA, USA.

Johann Roturier. 2009. Deploying Novel MT Technology to Raise the Bar for Quality: A Review of Key Advantages and Challenges. In *Proceedings of the twelfth Machine Translation Summit*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007a. Statistical Phrase-based Post-Editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, New York.

Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007b. Rule-based Translation With Statistical Phrase-based Post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206.

TAUS/CNGL Report. 2010. Machine Translation Post-Editing Guidelines Published. Technical report, TAUS.

Dusan Varis and Ondřej Bojar. 2017. CUNI System for WMT17 Automatic Post-Editing Task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 661–666, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

# DFKI-MLT System Description for the WMT18 Automatic Post-editing Task

**Daria Pylypenko**
DFKI Saarbrücken, Germany
daria.pylypenko@dfki.de

**Raphael Rubino**
DFKI Saarbrücken, Germany
raphael.rubino@dfki.de

## Abstract

This paper presents the Automatic Post-editing (APE) systems submitted by the DFKI-MLT group to the WMT'18 APE shared task. Three monolingual neural sequence-to-sequence APE systems were trained using target-language data only: one using an attentional recurrent neural network architecture and two using the attention-only (*transformer*) architecture. The training data was composed of machine translated (MT) output used as source to the APE model aligned with their manually post-edited version or reference translation as target. We made use of the provided training sets only and trained APE models applicable to phrase-based and neural MT outputs. Results show better performances reached by the attention-only model over the recurrent one, significant improvement over the baseline when post-editing phrase-based MT output but degradation when applied to neural MT output.

## 1 Introduction

For the 2018 edition of the WMT automatic post-editing (APE) task, two novelties were added compared to the previous editions: post-editing of neural machine translation (NMT) output in addition to phrase-based (PBMT) output, and the availability of larger training sets.

The DFKI-MLT systems developed for this shared task aimed at handling outputs from PBMT and NMT jointly with a single APE model. This was achieved by using artificial tokens indicating which type of MT system was used to produce the source segment and from which corpus the segment pair was extracted (inspired by (Yamagishi et al., 2016; Sennrich et al., 2016a; Johnson et al., 2017)).

Two NMT architectures were used to train our APE models, one using gated recurrent layers with global attention (Bahdanau et al., 2014), and one using attention and feed-forward layers without recurrence (Vaswani et al., 2017). The training data was composed of the official training set released by the shared task organizers plus subsets of the two additional resources filtered with bilingual cross-entropy difference (Axelrod et al., 2011).

The NMT architectures are described in Section 2 and the data preparation process is presented in Section 3. The results obtained by our APE models are compared to the baseline in Section 4. Finally, a conclusion is given in Section 5.

## 2 APE Architectures

The two neural network architectures used in our experiments were an attentional recurrent neural network with gated units and a multi-head attention-only network.

### 2.1 Recurrent Neural Network

For the Recurrent Neural Network (RNN) approach, we followed the architecture presented in (Bahdanau et al., 2014) and implemented in OpenNMT (Klein et al., 2017)[1]. Both the encoder and the decoder were 2-layered mono-directional RNNs with LSTM cells. The decoder applies global attention over the source sentence and performs input feeding. The source and target word embeddings, as well as the hidden layers, had 500 dimensions. The dropout probability was set to 0.3. The source and target vocabulary size is limited to 50000 tokens. Standard stochastic gradient descent is used as optimizer with a maximum batch size of 64. These hyper-parameters are the default ones in OpenNMT and were not tuned during the experiments presented in this paper.

[1] We used the Torch version of OpenNMT available at https://github.com/OpenNMT/OpenNMT

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 836–839
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64096

### 2.2 Attention Only

For the attention only approach, we used the architecture described in (Vaswani et al., 2017) and implemented in MARIAN (Junczys-Dowmunt et al., 2018). Two models were trained following this approach with variations in the number of heads (parallel attention layers), using 4 heads and 1024 dimensions for the feed-forward layers for one configuration (noted *Transformer small*) and 8 heads and 2048 dimensions for the second configuration (noted *Transformer large*). For both configurations, 512 dimensions were used for the embedding layers and the positional encodings, the dropout rate was set to 0.1 and the batch size to 32. These hyper-parameters were selected in order to compare the impact of increasing the dimensionality of the encoder and decoder layers, as well as the number of heads, on the post-editing performances.

## 3 Data Preparation

The training corpora provided for the APE shared task since 2016 were used (Bojar et al., 2016, 2017), as well as the two additional resources made available by the shared task organizers, namely the *artificial training data* presented in (Junczys-Dowmunt and Grundkiewicz, 2016) and the *eSCAPE* corpus (Negri et al., 2018). The target language data (German) was used for both input and output sequences in our APE models, the machine translated text being the source sequences and the corresponding post-edited text the target sequences, without making use of the source language (English). We did not split the machine translated data whether it was produced by a phrase-based (PBMT) or a neural (NMT) system. Instead, we added a specific token at the beginning of every source (machine translated) segment indicating which type of translation system was used to produce it.

The two additional parallel resources (*artificial training data* and *eSCAPE* corpora) were filtered using the bilingual cross-entropy difference approach presented in (Axelrod et al., 2011). We used the APE training data as in-domain corpus and each additional parallel corpus individually as out-of-domain corpus. The top $n$ sentence pairs ranked by their bilingual cross-entropy scores were kept, with $n$ being set by calculating the perplexity obtained on the development set. The resulting corpora used contain approx. 100k, 300k and 360k segment pairs taken from the *eSCAPE* PBMT corpus, the *eSCAPE* NMT corpus and the *artificial training data* respectively. Finally, we added a specific token at the beginning of every source segment indicating from which source it comes from: *eSCAPE*, *artificial* and *wmt*. The latter token was added to the official training data provided for the APE task, and to the development and test sets as well.

All datasets were used together to train our APE models, the artificial tokens inspired by (Yamagishi et al., 2016; Sennrich et al., 2016a; Johnson et al., 2017) allowed for identification of the segment pairs provenance. In order to balance the amount of data coming from different sources, we oversampled the official training data to reach approximately the amount taken from the two additional resources. Similarly, we increased the amount of data produced by a NMT system to balance with the amount produced by a PBMT system. This method was inspired by the work presented in (Chu et al., 2017).

The corpora which were not already tokenized were processed with the tokenizer distributed with the MOSES toolkit (Koehn et al., 2007). Additionally, all corpora were true-cased using a pre-trained true-casing model provided by the WMT organizers[2]. Finally, a byte-pair encoding (Sennrich et al., 2016b) model was trained on the German training data available for the WMT translation task and applied to both source and target sides of all corpora used in our experiments.

## 4 Evaluation

The three APE models trained for the shared task were used to post-edit the test set released by the organizers. Automatic evaluation with BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) was conducted by the organizers and the obtained scores on the official test set are reported in Table 1. The automatic metrics results are obtained by comparing each system output to the manually post-edited MT output (TER$_{pe}$ and BLEU$_{pe}$), to an independent translation (TER$_{ref}$ and BLEU$_{ref}$) and finally using both post-edited MT output and independent translation simultaneously as a multi-reference evaluation approach (TER$_{pe+ref}$ and BLEU$_{pe+ref}$). The results obtained by the non-post-edited MT output is presented as a baseline.

[2] http://data.statmt.org/wmt18/translation-task/preprocessed/de-en/

| System | TER$_{pe}$ | BLEU$_{pe}$ | TER$_{ref}$ | BLEU$_{ref}$ | TER$_{pe+ref}$ | BLEU$_{pe+ref}$ |
|---|---|---|---|---|---|---|
| | | | *PBMT Output* | | | |
| Baseline | 24.24 | 62.99 | 48.33 | 36.42 | 23.76 | 66.21 |
| Transformer large | 24.19 | **63.40** | **47.98** | **36.81** | 23.68 | **66.66** |
| Transformer small | 24.50 | 62.78 | 48.27 | 36.61 | 24.04 | 66.11 |
| RNN | 25.30 | 62.10 | 48.55 | 36.19 | 24.74 | 65.33 |
| | | | *NMT Output* | | | |
| Baseline | 16.84 | 74.73 | 42.24 | 44.22 | 16.27 | 76.83 |
| Transformer large | 18.86 | 70.98 | 43.74 | 41.53 | 18.37 | 72.93 |
| Transformer small | 18.84 | 70.87 | 43.79 | 41.53 | 18.41 | 72.95 |
| RNN | 19.88 | 69.35 | 44.28 | 40.91 | 19.43 | 71.36 |

Table 1: Automatic metrics results on the test set obtained by our APE models and compared to the baseline using three evaluation methods. Result in bold indicates significant improvement over the baseline.

The automatic evaluation results show that our models significantly degrades the baseline for the NMT output experiments when using the manually post-edited MT output, the independent translation and both simultaneously as gold reference to compute the scores. For the PBMT experiments, the model noted *Transformer large* significantly improves the PBMT output according to the BLEU metric for the three evaluation methods ($+0.4$pt for the post-edited MT output, $+.39$pt of the reference and $+.45$pt for both). However, the TER metric does not indicate significant improvements over the baseline when using the manually post-edited MT output as a gold reference.[3]

The degradation of NMT output in terms of automatic metrics might have at least two explanations. First, the lower amount of available training data produced by this type of MT system and provided by the organizers ($17,753$ unique tokens for NMT and $22,578$ for PBMT after truecasing). We used the over-sampling technique to balance the amount of NMT and PBMT data but this method does not increase the vocabulary coverage. Second, the baseline performances as indicated by the BLEU metric, $74.73$ and $44.22$ for the post-edited MT output and translation reference used as gold target respectively, are higher than the ones obtained with the PBMT experiments, which might be harder to outperform.

## 5 Conclusion

This paper presented the DFKI-MLT submissions to the WMT'18 APE shared task, which involved datasets produced by NMT and PBMT systems, as well as larger training data provided by the organizers. We evaluated two different APE architectures based on neural networks and made use of data preprocessing techniques to allow single models to be trained while being able to post-edit both NMT and PBMT outputs and using the target language data only.

The results as indicated by the BLEU metric showed that our approach brings significant improvement over the non post-edited PBMT output when using various gold references to compute the evaluation scores, but fails at improving NMT output. This might be due to the lower amount of training data produced by an NMT system compared to the PBMT produced data, and to the high performance reached by the baseline system on the NMT output as indicated by BLEU.

From the two APE architectures evaluated in our experiments and according to the automatic metrics used, the attention-only model outperformed the gated recurrent one for both types of MT output to post-edit. Both NN architectures could possibly reach better post-editing performances with careful hyper-parameters tuning and we plan to conduct these experiments in the future.

## Acknowledgments

The research reported in this paper is partially funded by the German Federal Ministry of Education and Research (BMBF) under the funding code 01IW17001 (Deeplee) and by the German Research Foundation (DFG) as part of SFB 1102 "Information Density and Linguistic Encoding". Responsibility for the content of this publication is with the authors.

[3] Significance tests were performed by the shared task organizers, more details are available in (Chatterjee et al., 2018).

## References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing*, pages 355–362.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 Shared Task on Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 385–391.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5(1):339–351.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 751–758.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. 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.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. escape: a large-scale synthetic corpus for automatic post-editing. *arXiv preprint arXiv:1803.07274*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the voice of a sentence in japanese-to-english neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210.

# Multi-encoder Transformer Network for Automatic Post-Editing

**Jaehun Shin and Jong-hyeok Lee**
**Department of Computer Science and Engineering,**
**Pohang University of Science and Technology**
**{jaehun.shin, jhlee}@postech.ac.kr**

## Abstract

This paper describes the POSTECH's submission to the WMT 2018 shared task on Automatic Post-Editing (APE). We propose a new neural end-to-end post-editing model based on the transformer network. We modified the encoder-decoder attention to reflect the relation between the machine translation output, the source and the post-edited translation in APE problem. Experiments on WMT17 English-German APE data set show an improvement in both TER and BLEU score over the best result of WMT17 APE shared task. Our primary submission achieves -4.52 TER and +6.81 BLEU score on PBSMT task and -0.13 TER and +0.40 BLEU score for NMT task compare to the baseline.

## 1 Introduction

Although machine translation technology has improved, machine translation output inevitably involves errors and the type of errors in the output varies depending on the machine translation system. Correcting those systematic errors inside the system may cause other problems such as increase of the decoding complexity (Chatterjee et al., 2015). For this reason, Automatic Post-Editing (APE) is suggested as an alternative to enhance the performance of the machine translation.

APE aims at the automatic correction of systematic errors in the machine translation output without any modification of the original machine translation system (Bojar et al, 2015; Bojar et al, 2016; Bojar et al, 2017). Basically, APE problem can be defined as a translation problem from machine translation output (*mt*) to post-edited sentence (*pe*), but source sentence (*src*) is used as an additional source for the problem. As a result, APE problem becomes a multi-source translation problem between two sources (*mt*, *src*) and a target (*pe*).

Due to the additional source, APE has two translation directions, the *mt*→*pe* direction and the *src*→*pe* direction. Previous researches have suggested various methods to combine the two directions with neural network architecture, such as log-linear combination of two translation models (Junczys-Dowmunt and Grundkiewicz, 2016), factored translation model (Hokamp, 2017) and multi-encoder architecture (Libovický et al., 2016; Chatterjee et al., 2017; Junczys-Dowmunt and Grundkiewicz, 2017; Variš and Bojar, 2017).

Among the methods, we focus on the multi-encoder approach because it is more appropriate to model the multi-source translation problem. Also, considering the importance of proper attention mechanism, as shown in the research of Junczys-Dowmunt and Grundkiewicz (2017), we use the transformer network (Vaswani et al., 2017) composed of a novel attention mechanism.

With this consideration, our submission to the WMT 2018 shared task on Automatic Post-Editing is a neural multi-encoder model based on the transformer network. We extend the transformer network implementation in Tensor2Tensor (Vaswani et al., 2018) library to implement our model. We participated in both PBSMT task and NMT task with this multi-encoder model.

In this paper, we introduce the multi-encoder transformer network for APE. The remainder of the paper is organized as follows: Section 2 contains the related work. Section 3 describes our method. Section 4 gives the experimental results, and Section 5 is the conclusion.

## 2 Related Work

### 2.1 Multi-Encoder Architecture

For a multi-source translation problem, the proper modeling of the relation between the multiple sources and the target is important. Combining two separate single-source translation models for

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 840–845
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64097

each source-target relation (Junczys-Dowmunt and Grundkiewicz, 2016) or constructing single input by combining the all sources (Hokamp, 2017) may be a solution, but these are not the exactly modeling the multi-source translation problem.

Zoph and Knight (2016) proposed the basic model of the multi-source translation problem. Their multi-encoder architecture uses trilingual data and contains separate encoders for each input to model the conditional probability of the target over the two sources. Libovický et al. (2016) showed the application of this multi-encoder architecture to model APE problem. They used the same architecture in both APE task and multi-modal translation task, because the two tasks can be defined as multi-source translation problem.

Although their model did not show a good result in the competition, the idea of multi-encoder architecture succeeded in the following WMT evaluation (Chatterjee et al., 2017; Junczys-Dowmunt and Grundkiewicz, 2017; Variš and Bojar., 2017) and achieved good results.

### 2.2 Transformer Network

Transformer network is a novel neural machine translation architecture proposed by Vaswani et al. (2017), which avoids recurrence and convolution and focuses on the attention mechanism. The network utilizes an encoder-decoder architecture based on the stacked layers and each layer uses a new novel attention mechanism called multi-head attention.

Multi-head attention is a variation of scaled dot-product attention. It employs a number of attention heads for information from different representation subspaces at different positions. With this characteristic, multi-head attention can model the dependency between tokens regardless of their distance up to the number of heads.

Transformer network uses the multi-head attention in three different ways: self-attention in encoder, masked self-attention in decoder, and encoder-decoder attention. The self-attention and the masked self-attention model the internal dependency of the input and the output respectively, and the encoder-decoder attention models the dependency between the input and the output.

With this attention mechanism, transformer network achieved the state-of-the-art result on the WMT 2014 English-to-German and English-to-French translation tasks, and were faster to train than other prior models (Vaswani et al., 2017).

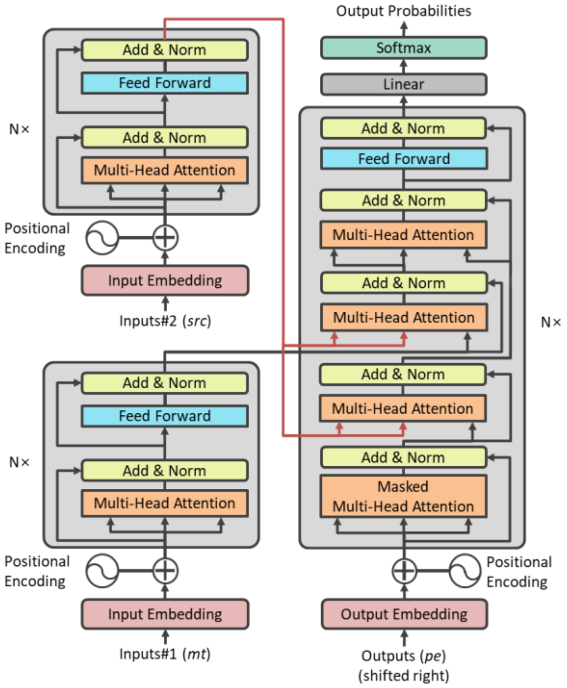


Figure 1: The overall architecture of multi-encoder transformer network for automatic post-editing task.

## 3 Multi-Encoder Transformer Network

In a normal multi-source translation problem, all of the sources and the target are assumed to be a different representation of a common abstracted meaning. However, in APE problem, we cannot adopt this assumption because the machine translation output is considered to have systematic errors. These errors make a gap between the machine translation output and the post-edited sentence. Therefore, for APE problem, we should aim to reduce the gap, not to find the common abstracted meaning. In this intuition, the three directions should be considered to model the APE problem, sentence correction (*mt*→*pe*), ideal translation (*src*→*pe*), and original translation (*src*→*mt*).

Even though Bérard et al. (2017) used a chained architecture for the context information of original translation, most of previous approaches focused on combining sentence correction and ideal translation. However, in terms of reducing the gap, APE problem is close to modeling the relation between original translation and ideal translation, rather than the relation between the machine translation output and the post-edited sentence.

Our multi-encoder transformer network is based on this idea. Figure 1 illustrates the overall architecture of our multi-encoder transformer network for APE problem. We extend transformer network to have two encoders, one for the machine translation output and the other for the source sentence. Each encoder has its own self-attention layer and feed-forward layer to process each input separately. Also, we add two multi-head attention layers to decoder, one for original translation dependency (*src→mt*) and another for ideal translation dependency (*src→pe*). After these attention layers, the words common to both the machine translation output and the post-edited sentence have similar dependency on the source sentence, so those common words obtain similar source contexts. Then we apply multi-head attention between the output of those attention layers, expecting that the source context helps the decoder to recognize those common words which should be remained in post-edited sentence.

In short, we added the second encoder for the source sentence to the transformer network and modified the encoder-decoder attention structure to reflect the relation between the original translation and the ideal translation.

# 4 Experimental Results

## 4.1 Data

We used WMT'18 official data set (Chatterjee et al., 2018) for PBSMT task and NMT task individually. The official PBSMT data set consists of training data, development data and two test data (2016, 2017), and the official NMT dataset consists of training data and development data.

We adopted the artificial training data (Junczys-Dowmunt and Grundkiewicz, 2016) as an additional training data for both tasks. Table 1 summarizes the statistic of the data sets. In addition, the artificial-small data set is the subset of the artificial-large data set.

## 4.2 Training Parameters

We used the base model parameters of transformer network: 6 stacks, 8 heads, 512 hidden dimension, 2,048 feed-forward dimension, 64 key dimension, 64 value dimension, dropout probabilities 0.1 and Adam optimization with $\beta_1$=0.9, $\beta_2$=0.997 and $\varepsilon$=$10^{-9}$.

We built a shared word piece vocabulary with size of $2^{16}$ from the combined set of PBSMT training data set and artificial-large data set for PBSMT model. For NMT model, we used the combined set of official data and artificial-small data to build the vocabulary, with consideration of the difference between two tasks.

| Task | Data set | Sentences | TER |
|---|---|---|---|
| PBSMT | training set | 23,000 | 25.35 |
| | development set | 1,000 | 24.81 |
| | test set 2016 | 2,000 | 24.76 |
| | test set 2017 | 2,000 | 24.48 |
| | artificial-small | 526,368 | 25.55 |
| | artificial-large | 4,391,180 | 35.37 |
| NMT | training set | 13,442 | 14.89 |
| | development set | 1,000 | 15.08 |

Table 1: Statistics for WMT APE data sets.

For training, we used a mini batch size of 2,048 with max sequence length of 256 and initial learning rate of 0.2. We set warmup steps to 16k and trained the model during 160k steps. Model checkpoints were saved every 1,000 mini batches. We select this model as our base model.

## 4.3 Tuning

After 160k steps of training, we tuned the base model in two step. For the first tuning step, we reduced the training data to the sum of the official training data set and artificial-small data set. We trained the base model on the reduced training data during 30k steps more and selected the model with the lowest validation loss (1st-tuned).

For the second tuning step, we used the official training data to fine-tune the 1st-tuned model. We used the same tuning method with 1k training step. The model with lowest validation was selected as the final model (2nd-tuned).

## 4.4 Evaluation

We evaluated the models using the WMT data set, computing the TER (Snover et al., 2006) and BLEU (Papineni et al., 2002) scores on the decoded output. The decoding parameter is the same as the default decoding parameter of the Tensor2tensor. We used the scores of original machine translation output as the baseline to compare our results. Table 2 shows the results of the evaluation on PBSMT data set and NMT data set.

The result on PBSMT data set is comparable to the last year's top result without any additional post-processing. In contrast, the result on NMT data set shows almost no improvement. We guess that the different characteristics of PBSMT artificial data set from the NMT training data set causes the result.

| model | PBSMT | | | | | | NMT | |
|---|---|---|---|---|---|---|---|---|
| | dev | | test 2016 | | test 2017 | | dev | |
| | **TER**↓ | BLEU↑ | **TER**↓ | BLEU↑ | **TER**↓ | BLEU↑ | **TER**↓ | BLEU↑ |
| MT Baseline | 24.81 | 62.92 | 24.76 | 62.11 | 24.48 | 62.49 | **15.08** | 76.76 |
| Multi-T2T_base | 22.80 | 66.36 | 22.70 | 65.84 | 22.98 | 65.46 | 16.73 | 74.43 |
| Multi-T2T_1$^{st}$-tuned | 21.11 | 68.78 | 21.20 | 67.95 | 21.64 | 67.33 | 15.76 | 76.02 |
| Multi-T2T_2$^{nd}$-tuned | **19.05** | 71.79 | **19.14** | **70.98** | **19.26** | **70.50** | 15.27 | **76.88** |
| Chatterjee et al. (2017)* | 19.22 | **71.89** | 19.32 | 70.88 | 19.60 | 70.07 | — | — |

Table 2: The result of multi-encoder transformer network on WMT APE data set.

| model | PBSMT | | | | | | NMT | |
|---|---|---|---|---|---|---|---|---|
| | dev | | test 2016 | | test 2017 | | dev | |
| | **TER**↓ | BLEU↑ | **TER**↓ | BLEU↑ | **TER**↓ | BLEU↑ | **TER**↓ | BLEU↑ |
| Mutli-T2T_top5-avg | 18.87 | 71.72 | **19.15** | **70.88** | **18.82** | **70.86** | 14.97 | 77.22 |
| Mutli-T2T_fix5-avg | 18.88 | 71.68 | 19.22 | 70.80 | 18.90 | 70.78 | 14.96 | 77.25 |
| Mutli-T2T_var5-avg | **18.85** | **71.83** | 19.19 | 70.75 | 18.85 | 70.68 | 14.97 | 77.25 |
| Mutli-T2T_top1 | 18.91 | 71.66 | 19.23 | 70.78 | 18.91 | 70.74 | **14.94** | **77.26** |

Table 3: The results of submitted models on WMT APE data set.

| Task | Systems | **TER**↓ | BLEU↑ |
|---|---|---|---|
| PBSMT | WMT18-Baseline | 24.24 | 62.99 |
| | PRIMARY (top5) | 19.72 | 69.80 |
| | CONTRASTIVE1 (fix5) | **19.63** | **69.87** |
| | CONTRASTIVE2 (var5) | 19.74 | 69.70 |
| NMT | WMT18-Baseline | 16.84 | 74.73 |
| | PRIMARY (fix5) | 16.71 | 75.13 |
| | CONTRASTIVE1 (top1) | **16.70** | 75.14 |
| | CONTRASTIVE2 (var5) | 16.71 | **75.20** |

Table 4: The official results of the submitted models to WMT18 APE task..

### 4.5 Submitted System

We used checkpoint averaging to make an ensemble model for submission candidates. For the better result, we used various checkpoint saving frequencies in the second tuning step and trained the model five times for each frequency. Then, we applied checkpoint averaging on the models with following conditions: top-5 models (top5), top-5 models in a fixed checkpoint frequency (fix5), five top-1 models for various checkpoint frequencies (var5). We used TER score on the development data set to select the models. In addition, we chose the top-1 model to the submission candidate. Table 3 summarizes the result of the four submission candidates on both PBSMT and NMT data set. For the submission, we chose three models with low TER score and high BLEU score.

Table 4 shows the official result of the submitted model on WMT18 test data set. Our primary submission for PBSMT achieves -4.52 TER and +6.81 BLEU scores and our primary submission on NMT task -0.13 TER and +0.40 BLEU scores compare to the baseline.

## 5 Conclusion

In this paper, we propose a multi-encoder transformer network for APE task. We modified the structure of encoder-decoder attention to reflect the relation between machine translation output, source sentence and post-edited sentence in APE. Our multi-encoder model showed a comparable result to the top result of last year's competition on PBSMT task, although almost no improvement on NMT task.

## Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (R7119-16-1001, Core technology development of the real-time simultaneous speech translation based on knowledge enhancement).

## References

Alexandre Bérard, Laurent Besacier, and Olivier Pietquin. 2017. LIG-CRIStAL Submission for the WMT 2017 Automatic Post-Editing Task. In *Proceedings of the Second Conference on Machine Translation*, pages 623-629.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Lisbon, Portugal, pages 1-46.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 131–198.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Vavara Logacheva Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, pages 169-214.

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the Planet of the APEs: a Comparative Study of State-of-the-art Methods for MT Automatic Post-Editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156-161.

Rajen Chatterjee, Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source Neural Automatic Post-Editing: FBK's participation in the WMT 2017 APE shared task. In *Proceedings of the Second Conference on Machine Translation (Volume 2: Shared Task Papers)*, pages 630-638.

Rajen Chatterjee, Matteo Negri, Raphael Rubino and Marco Turchi. 2018. Findings of the WMT 2018 Shared Task on Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics, Brussels, Belgium.

Chris Hokamp. 2017. Ensembling Factored Neural Machine Translation Models for Automatic Post-Editing and Quality Estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 647-654.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 751–758.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. The AMU-UEdin Submission to the WMT 2017 Shared Task on Automatic Post-Editing. In *Proceedings of the Second Conference on Machine Translation,* pages 639-646.

Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Ondřej Bojar, and Pavel Pecina. 2016. CUNI system for wmt16 automatic post-editing and multimodal translation tasks. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 646-654.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311-318

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*. Vol. 200, No. 6

Dušan Variš and Ondřej Bojar. 2017. CUNI System for WMT17 Automatic Post-Editing Task. In *Proceedings of the Second Conference on Machine Translation*, pages 661-666.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser and Illia Polosukhin. 2017. Attention is all It shows that our multi-encoder model has a sufficient potential to solve APE problem.you need. In *Advances in Neural Information Processing Systems*, pages 5998-6008.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*. https://arxiv.org/abs/1803.07416

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. CoRR abs/1601.00710. http://arxiv.org/abs/1601.00710.

# Multi-source Transformer with Combined Losses for Automatic Post-Editing

**Amirhossein Tebbifakhr**[1,2], **Ruchit Agrawal**[1,2], **Matteo Negri**[1], **Marco Turchi**[1]
[1] Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento - Italy
[2] University of Trento, Italy
{atebbifakhr,ragrawal,negri,turchi}@fbk.eu

## Abstract

Recent approaches to the Automatic Post-editing (APE) of Machine Translation (MT) have shown that best results are obtained by neural multi-source models that correct the raw MT output by also considering information from the corresponding source sentence. To this aim, we present for the first time a neural multi-source APE model based on the Transformer architecture. Moreover, we employ sequence-level loss functions in order to avoid exposure bias during training and to be consistent with the automatic evaluation metrics used for the task. These are the main features of our submissions to the WMT 2018 APE shared task, where we participated both in the PBSMT subtask (i.e. the correction of MT outputs from a phrase-based system) and in the NMT subtask (i.e. the correction of neural outputs). In the first subtask, our system improves over the baseline up to -5.3 TER and +8.23 BLEU points ranking second out of 11 submitted runs. In the second one, characterized by the higher quality of the initial translations, we report lower but statistically significant gains (up to -0.38 TER and +0.8 BLEU), ranking first out of 10 submissions.

## 1 Introduction

The purpose of Automatic Post-Editing (APE) is to correct the raw output of a Machine Translation system by learning from human corrections. Since the inner workings of MT engines are often not accessible (e.g. by users relying on Google Translate), hence impossible to modify and improve, APE becomes a solution to enhance the quality of the translated segments. Good solutions to the problem have high potential in the translation industry, where better translation means lower costs for human revision and where the adaptations of third-party, general-purpose systems to new projects is a major need.

In the last few years, the APE shared tasks at WMT (Bojar et al., 2015, 2016, 2017) have renewed the interests in this topic and boosted the technology around it. Moving from the phrase-based approaches used in the first editions of the task (Chatterjee et al., 2015), last year the multi-source neural models (Chatterjee et al., 2017; Junczys-Dowmunt and Grundkiewicz, 2017; Hokamp, 2017) have shown their capability to significantly improve the output of a PBSMT system. These APE systems shared several features and implementation choices, namely: *1)* an RNN-based architecture, *2)* the use of large artificial corpora for training, *3)* model ensembling techniques, *4)* parameter optimization based on Maximum Likelihood Estimation (MLE) and *5)* vocabulary reduction using the Byte Pair Encoding (BPE) technique. Although they achieve good performance and impressive translation quality improvements, some of these techniques are not optimal for the actual deployment of APE technology in the translation industry. The main reasons are the long time required for model training and the high maintenance costs of complex architectures that combine multiple models. To make APE solutions usable and useful for the industrial market, our submissions focus on the development of an end-to-end system that does not require multiple models and external components (e.g. hypothesis re-ranker), but leverages a fast to train architecture, effective pre-processing methods and task-specific losses to boost performance. Our main contributions are:

- We adapt the Transformer (Vaswani et al., 2017) to the APE problem, so that multiple encoders can be exploited to leverage information both from the MT output to be corrected and from the corresponding source sentence (multi-source encoding).

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 846–852
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64098

- We explore different strategies for combining token and sentence level losses.
- We apply *ad hoc* pre-processing for the German language by re-implementing the pipeline used by the best system at the WMT‘17 Translation task (Huck et al., 2017).
- In addition to the artificial data released by (Junczys-Dowmunt and Grundkiewicz, 2016), we make extensive use of a synthetic corpus of 7.2M English-German triplets (Negri et al., 2018), which was provided by the organizers as additional training material.

We participated in both the APE‘18 subtasks with positive results. In the PBSMT subtask our top run improves the baseline up to -5.3 TER and +8.23 BLEU points (ranking second out of 11 submissions) while, in the NMT subtask, it achieves a -0.38 TER and +0.8 BLEU improvement (ranking first out of 10 submissions).

## 2 Multi-source Transformer Network

The Transformer network (Vaswani et al., 2017), like most of the sequence-to-sequence models, follows an encoder-decoder architecture. It uses stacked layers for the encoder and the decoder. The encoder layers consist of a multi-head self-attention, followed by a position-wise feed-forward network. The decoder layers have an extra multi-head encoder-decoder attention after the multi-head self-attention sub-layer. Also, a softmax normalization is applied to the output of the last layer in the decoder to generate a probability distribution over the target vocabulary. Since there is no recurrence in this architecture, a positional encoding is added to both the source and the target word embeddings in order to empower the model to capture the position of the words. More formally, the positional encoding is defined as follows:

$$\text{PE}(pos, 2i) = \sin(pos/10000^{2i/d_{model}})$$
$$\text{PE}(pos, 2i+1) = \cos(pos/10000^{2i/d_{model}})$$

where $pos$ is the position of the word in the sentence, $i$ is the dimension of the vector, and $d_{model}$ is the dimensionality of the word embeddings.

The attention is a mapping from a query ($Q$), a key ($K$), and a value ($V$) to an output vector. In Transformer, the attention is based on dot-product attention which is defined as follow:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d_k})V$$

where $d_k$ is added as a scaling factor for improving the numerical stability, which is equal to the dimensionality of the key matrix. The multi-head attention receives $h$ different representations of $(Q, K, V)$, which makes it possible to learn different relationships between information coming from different positions simultaneously. It is computed as follows:

$$\text{MH}(Q, K, V) = \text{Concat}(head_1, ..., head_h)W^O$$

where $h$ is number of heads and $W^O$ is a parameter matrix with $hd_v{*}d_{model}$ dimension. In Transformer, the multi-head attention is used in two different ways: encoder-decoder and self-attention. In the self-attention, in both the encoder and the decoder, the $Q$, $K$, and $V$ matrices are coming from the previous layer, while in the encoder-decoder attention, $Q$ matrix comes from the previous layer, and the $K$ and $V$ matrices come from the encoder.

In order to encode the source sentence in addition to the MT output, we employ the multi-source method by Zoph and Knight (2016). Our model consists of two encoders, one for the source sentence and one for the MT output. The outputs of these two encoders are concatenated and passed as the key in the attention. This helps to have a better representation, leading to a more effective attention at decoding time.

## 3 Sequence-Level Loss Function

For training the model, most of the approaches in sequence-to-sequence modeling try to maximize the likelihood over the training data. In this scenario, the loss function is a token-level loss defined as:

$$\mathcal{L}_{\text{MLE}} = -\sum_{n=1}^{N} p(y_n|y_{<n}, \mathbf{x})$$

where $p(y_n|y_{<n}, \mathbf{x})$ is the probability of generating the target word in the $n$-th position. Ranzato et al. (2015), however, indicate two drawbacks for Maximum Likelihood Estimation (MLE). First, during training, the previous words passed to the decoder are always chosen from the ground-truth. However, the fact that at test time the previous

words are chosen from the model distribution, results in a bias called *exposure bias*. Such bias makes the model unable to recover from the errors made in the decoding step, which easily have a cumulative catastrophic effect. Second, using MLE as loss function, the model is optimized to maximize the probability of the training data, while the performance of the model is evaluated by the sequence-level evaluation metrics (TER and BLEU in the case of the APE task). In order to overcome the mentioned drawbacks, following Minimum Risk Training (MRT) introduced by Shen et al. (2016), we use a risk function which is defined as:

$$\mathcal{R}_{\text{MRT}} = \sum_{\mathbf{y}\in\mathcal{S}(\mathbf{x})} \frac{P(\mathbf{y}|\mathbf{x})}{\sum_{\mathbf{y}'\in\mathcal{S}(\mathbf{x})} P(\mathbf{y}'|\mathbf{x})} \Delta(\mathbf{y})$$

where $\mathcal{S}(\mathbf{x})$ is a set of sampled hypotheses from the model for the input sentence $\mathbf{x}$, $P(\mathbf{y}|\mathbf{x})$ is the probability of the sampled hypothesis, and $\Delta(\mathbf{y})$ is a cost value for generating the sample $\mathbf{y}$, e.g. $\Delta(\mathbf{y}) = -\text{BLEU}(\mathbf{y})$. Following Sennrich et al. (2016), we employ negative smoothed sentence-level BLEU (Papineni et al., 2002; Chen and Cherry, 2014) for computing the cost function.[1]

## 4 Data Pre-processing

In order to reduce the vocabulary size, on the German MT output and post-edits we apply our reimplementation of the word segmentation method introduced by Huck et al. (2017). It consists of three different steps:

1. Suffixes are separated from the word stems by using a modified version of snowball stemming, which separates and keeps the suffixes instead of stripping them;
2. The output of the previous step is passed to the empirical compound splitter described in (Koehn and Knight, 2003), which is run with the same parameters reported in (Huck et al., 2017);
3. The output of the previous step is segmented with Byte Pair Encoding (BPE) (Sennrich et al., 2016).

For the English source sentences, we only use BPE to reduce the vocabulary size.

## 5 Experimental Setting

### 5.1 Data

To train our models, we used both the in-domain data released by APE the task organizers and the synthetic data provided as additional training material.

**In-domain Data.** In-domain data consist of English-German (SRC, MT, PE) triplets in which the MT element (a German translation of the English SRC sentence) has been generated by "black-box" MT systems: a phrase-based one for the PBSMT subtask and a neural one for the NMT subtask. In both cases, the post-edit element (PE) is a correction of the target made by professional post-editors. The PBSMT training set, which is the largest one, comprises 28K triplets. The NMT training set, is smaller in size and contains 13K instances. From the two training corpora, we extracted 1K triplets to be used as development set to compare the performance of different models during training.

**Synthetic data.** Since building neural APE models heavily relies on the availability of large training data, we took advantage of the following two corpora:

- the eSCAPE corpus (Negri et al., 2018), which contains 7.2M English-German triplets for each MT paradigm (i.e. 7.2M phrase-based and neural translations of the same source sentences). It has been generated from a parallel English-German corpus, by taking the target sentences as artificial post-edits and the machine-translated source sentences as MT elements of each triplet.
- The artificial corpus provided by Junczys-Dowmunt and Grundkiewicz (2016), which contains 4.0M English-German triplets generated by applying a round-trip translation protocol to German monolingual data.

Before applying the pre-processing described in Section 4 to the eSCAPE data, we performed the following two cleaning steps:

1. We removed the triplets in which the length ratio between the source sentence and the

[1] Although TER (Snover et al., 2006) is the primary evaluation metric for the task, we opted for BLEU since, according to (Shen et al., 2016), optimizing with this metric gives better results also when evaluation is done with TER.

post-edit is too different from the average in the corpus;

2. We run a language identifier[2] in order to remove the triplets having a non-English source sentence or a non-German post-edit.

The application of these two cleaning steps resulted in the removal of approximately 600K instances from the eSCAPE corpus.

### 5.2 Evaluation Metrics

In order to evaluate our models, we use the two automatic evaluation metrics: i) TER which is computed based on edit distance (Snover et al., 2006) and ii) BLEU which is the geometric mean of n-gram precisions multiplied to the brevity penalty (Papineni et al., 2002).

### 5.3 Hyperparameters

We set the number of merging rules to 32K for applying BPE in the pre-processing steps. We employ OpenNMT-tf toolkit (Klein et al., 2017) for our implementation, by using 512 dimensions for word embeddings, 4 layers for both the encoders and the decoder with 512 units, and feed-froward dimension of 1,024. In order to avoid over-fitting, we use attention and residual dropout by setting the dropout probability to 0.1, along with the label-smoothing with parameter equal to 0.1. For training using MLE, we use the Adam optimizer (Kingma and Ba, 2014) with batch size of 8,192 tokens, learning rate of 2.0 and the warm-up strategy introduced by (Vaswani et al., 2017) with the warm-up steps equals to 8,000. For training using MRT, we use stochastic gradient descent optimizer with the batch size of 4,096 tokens. We also employ the beam search with beam width of 5 to sample hypotheses from the model.

## 6 Results

For both the subtasks, we train six different models. The performance of these models on the PBSMT and NMT development sets is reported in Tables 1 and 2.

**Generic.** First, we train a model using the union of the (out-domain) synthetic datasets. As expected, the performance of this model in both subtasks is lower than the baseline. We only train this model as initial generic model in order to fine-tune it using the in-domain data.

**MLE.** Using MLE, we fine-tune the generic model on the corresponding in-domain data for each subtask. For the PBSMT subtask, this model achieves a -6.73 TER and +9.94 BLEU improvement over the baseline. The gain is much lower for the NMT subtask (-0.33 TER and +0.85 BLEU), confirming that, together with the availability of less training data, the quality of the underlying NMT system has left little space for improvement.

**MRT.** We continue the training by using MRT in two ways: *i)* by adding the reference to the set of hypotheses sampled from the model and *ii)* without adding the reference. In contrast with Shen et al. (2016), who suggest to add the reference to the sampled set of hypotheses, we found that adding the reference is harmful. Actually, by adding the reference to the sample, the other hypotheses are considered as poor alternatives, since they have a lower BLEU score. Nevertheless, these samples usually have good quality and a considerable overlap with the reference. Therefore, updating the model in the direction of decreasing the probability of these hypotheses is does not seem a promising direction.

**MRT + MLE.** In order to avoid this problem and take advantage of the reference, we re-run the previous learning step using the linear combination of the two loss functions. Formally, we use the following loss function:

$$\mathcal{L}_{comb} = \alpha\mathcal{L}_{\text{MLE}} + (1-\alpha)\mathcal{R}_{\text{MRT}}$$

where $\alpha$ is set to 0.5 to give equal importance to the two components[3] The results show that combining the two loss functions makes the model able to learn also from the reference without ignoring the contribution of the other hypotheses.

Our model outperforms the best performing system at the last round of the shared task (Chatterjee et al., 2017), with improvements up to -1.27 TER and +1.23 BLEU on the PBSMT development set. Although we are using more out-of-domain data, it is interesting to note that these scores are obtained with a much simpler architecture, which does not require to ensemble $n$ models and to train a re-ranker. Since using only

[2] For this purpose we used the language detector available at: https://github.com/optimaize/language-detector.

[3] We leave for future work the empirical estimation of optimal values for $\alpha$.

| Model | Reference | TER | BLEU |
|---|---|---|---|
| Generic | - | 25.00 | 61.69 |
| MLE | - | 18.08 | 72.86 |
| MRT | Yes | 18.02 | 72.91 |
| MRT | No | 18.44 | 73.05 |
| MLE + MRT | Yes | 17.99 | 72.99 |
| MLE + MRT | No | **17.95** | **73.12**$^1$ |

Table 1: Results of the multi-source Transformer with specific losses on the PBSMT outputs. The performance of the MT baseline are 24.81 TER and 62.92 BLEU. Superscript 1 denotes that improvement over MLE is statistically significant.

| Model | Reference | TER | BLEU |
|---|---|---|---|
| Generic | - | 17.35 | 72.55 |
| MLE | - | 14.75 | 77.61 |
| MRT | Yes | 14.81 | 77.57 |
| MRT | No | 14.78 | **77.74** |
| MRT + MLE | Yes | 14.75 | 77.68 |
| MRT + MLE | No | **14.68** | 77.68 |

Table 2: Results of the multi-source Transformer with specific losses on the NMT outputs. The performance of the MT baseline are 15.08 TER and 76.76 BLEU.

MRT produced a better BLEU score in NMT subtask, we submitted the best model using only MRT without reference as our *Primary* submission, and the best model using MRT+MLE as our *Contrastive* submission.

## 7 Test Set Results

The performance of the primary and contrastive APE systems on the test set for both the subtasks is reported in Table 3. Apart from a minimal variation in the TER scores for the PBSMT subtask, the results confirm what previously seen on the development set. Our APE systems are able to significantly improve the quality of the PBSMT outputs by achieving a gain of -5.62 TER and +8.23 BLEU points.

When post-editing the output of a NMT system, the gains are smaller (-0.38 TER and +0.8 BLEU). This is somehow expected since the role of an APE system is to fix MT errors: in presence of higher quality translations (from PBSMT to NMT: -7.4 TER and +12.54 BLEU) there are less errors to correct and the chance to apply unnecessary changes is higher. Apart from that, our APE systems are able to improve the NMT outputs showing that, even in this challenging condition, APE is useful.

| Task | System | TER | BLEU |
|---|---|---|---|
| PBSMT | Baseline | 24.24 | 62.99 |
| | Primary | 18.94 | **71.22** |
| | Contrastive | **18.62** | 71.04 |
| NMT | Baseline | 16.84 | 74.73 |
| | Primary | **16.46** | **75.53** |
| | Contrastive | 16.55 | 75.38 |

Table 3: Submissions at the WMT APE shared task.

## 8 Conclusion

We presented the FBK's submissions to the APE shared task at WMT 2018. Our models extend a Transformer-based architecture by: *1)* leveraging multi-source inputs consisting in the source and MT texts and *2)* taking advantage of combined token and task-specific losses. Moreover, an *ad hoc* text pre-processing for the German language and more artificial data are exploited to help the training of the model. The resulting systems show large gains in performance when post-editing the PBSMT translations (our top-submission ranks second in this subtask), while minimal improvements are obtained when correcting the NMT outputs (still, our top-run ranks first in this subtask). These differences in performance strongly depend on the initial quality of the MT outputs that significantly changes from the PBSMT to the NMT system.

It is worth to remark that our implementation choices were mainly driven by the needs of a translation market in which simple solutions that are easy to maintain are always preferable to complex architectures. In this direction, our APE systems consist of a single network that can be trained in an end-to-end fashion, without recourse to ensembles of multiple models or the concatenation of components (e.g. hypothesis re-ranker) that have to be trained independently.

## Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara

Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source neural automatic post-editing: Fbk's participation in the wmt 2017 ape shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 630–638. Association for Computational Linguistics.

Rajen Chatterjee, Marco Turchi, and Matteo Negri. 2015. The fbk participation in the wmt15 automatic post-editing shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 210–215, Lisbon, Portugal. Association for Computational Linguistics.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367.

Chris Hokamp. 2017. Ensembling factored neural machine translation models for automatic post-editing and quality estimation. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 647–654, Copenhagen, Denmark. Association for Computational Linguistics.

Matthias Huck, Fabienne Braune, and Alexander Fraser. 2017. Lmu munich's neural machine translation systems for news articles and health information texts. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 315–322, Copenhagen, Denmark. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. The amu-uedin submission to the wmt 2017 shared task on automatic post-editing. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 639–646, Copenhagen, Denmark. Association for Computational Linguistics.

D. P. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. *ArXiv e-prints*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 187–193, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. eSCAPE: a Large-scale Synthetic Corpus for Automatic Post-Editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. 2015. Sequence Level Training with Recurrent Neural Networks. *ArXiv e-prints*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. *arXiv preprint arXiv:1601.00710*.

# The Speechmatics Parallel Corpus Filtering System for WMT18

**Tom Ash, Remi Francis, Will Williams**
Speechmatics, Cambridge, United Kingdom
{toma,remi,willw}@speechmatics.com

## Abstract

Our entry to the parallel corpus filtering task uses a two-step strategy. The first step uses a series of pragmatic hard 'rules' to remove the worst example sentences. This first step reduces the effective corpus size down from the initial 1 billion to 160 million tokens. The second step uses four different heuristics weighted to produce a score that is then used for further filtering down to 100 or 10 million tokens. Our final system produces competitive results without requiring excessive fine tuning to the exact task or language pair. The first step in isolation provides a very fast filter that gives most of the gains of the final system.

## 1 Introduction

This task asks for applicants to provide a score for each sentence pair in a 1-billion-word Machine Translation (MT) training corpus that is considered to be 'very noisy', such that those scores can be used to filter the corpus down into 10 million and 100 million words subsets. The quality of the output is measured by BLEU score obtained by training standard systems on these two subsets of data.

We consider this task to comprise of two primary components, namely (a) removing sentences that do not represent good examples of translation from one language to the other ('junk') and (b) distilling the remaining data down to a smaller training footprint without losing quality or diversity and then attaching scores to those sentences.

These two components are somewhat related; however, we chose to use a two-pass system to tackle them independently, so our system could be used to tackle the two components separately if required by a 'real-world' use case.

There are various approaches to this task that have previously been reported and we have attempted to select the most pragmatically useful of these to incorporate into our final system. Our philosophy in choosing what to put into our system was to make it as general as possible, such that it could be used for other language pairs and different datasets, rather than specifically tuning for this task. That then allows us to use the system more widely across our efforts in the field of machine translation. We have also chosen to use an array of different metrics to produce a final score, rather than a single score, to gain the benefits of multiple models that approach the problem in different ways.

### 1.1 Dev Data

As well as the 1-billion-word corpus to be processed, a smaller corpus of paired English-German data is available as a development set. This data comprises the data for the WMT 2018 news translation task data for German-English without the Paracrawl parallel corpus. This data is approximately 130M words, drawn from Europarl, Common Crawl, News Commentary and Rapid EU Press Release Corpora. More details of this data are available from http://www.statmt.org/wmt18/translation-task.html.
This data is hereafter referred to as the 'dev data'.

## 2 System Description

Our filtering system consists of two passes. The first pass uses some hard 'rules' to eliminate the bulk of the data. We consider this data to be 'junk' and score each sentence thus removed with a zero.

The second pass uses several heuristics we have developed to assign scores greater than zero to

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 853–859
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64099

each sentence pair, with the aim of distilling down the data into as rich a subset as possible.

### 2.1 Initial 'rules'

The following hard rules are performed sequentially on the corpus. If any sentence 'fails' a rule it is immediately given a score of 0 and not considered for any further portion of our scoring system.

**Line Length:** we follow the 'length-based filtering' of Khadivi and Ney (2005). This method attempts to catch instances of grossly mistranslated sentences using the assumption that sentences in different languages will consist of approximately the same number of words and removing sentence pairs that have widely varying lengths.

If I and J denote the source and target sentence length respectively, sentence pairs are eliminated unless all of the following are true:

$$\begin{aligned}&6 * \mathrm{I} > \mathrm{J} \text{ and } \mathrm{I} < 6 * \mathrm{J}\\&\mathrm{I} < 3 \text{ or } \mathrm{J} < 3 \text{ or } (\mathrm{I} < 2.2 * \mathrm{J} \text{ and } \mathrm{J} < 2.2 * \mathrm{I})\\&\mathrm{I} < 10 \text{ or } \mathrm{J} < 10 \text{ or } (\mathrm{I} < 2 * \mathrm{J} \text{ and } \mathrm{J} < 2 * \mathrm{I}) \qquad (1)\end{aligned}$$

We sampled these same thresholds on a range of other languages and were surprised to see they were reasonable without alteration even in quite diverse situations, such as agglutinative languages.

**Non-translation:** following Song et. al. (2014) we remove sentence pairs where the source and target have a BLEU similarity score greater than 0.6. This deals with cases of either untranslated or only partially translated sentences.

**Language identification:** Web crawled corpora typically contain many data that are not in the language it claims to be. To try and identify such cases we use *lang-id* (Lui and Baldwin, 2012) to identify the most likely language of both the source and target sentence and remove the entry if either source or target disagrees with the correct label.

We also tried a different version of this in which we used the language probabilities generated by langid alongside a threshold instead of a binary decision based on the langid 1-best. With appropriate tuning this gave marginal gains, but the processing time was increased more than we found acceptable so is not used in our target system.

For languages not supported by pre-trained language identification models, we intend to use FastText (Joulin et. al, 2017) to train our own.

We believe this is the part of our rules most likely to give false positives. It was not possible to quantify this, but from qualitative judgement of the output it appeared to often falsely misjudge something as being in an incorrect language, particularly short sentences. Nonetheless our experiments show the rule greatly improved overall quality of the final corpus, so we believe it provides a lot more good than harm.

**Character filtering:** we expect there to be unwanted characters in a noisy corpus – for example Denkowski et. al. (2012) filter out all lines with invalid Unicode, control characters and similar. We approach this in a systematic way, by defining a list of characters we deem acceptable for each language and only keeping sentences containing just those characters. We create our character lists by counting character occurrence in the 'dev data', sorting on character count and then quickly manually scanning through the most common characters to generate a final list of around 80 characters per language that we deem 'acceptable'.

Our system then eliminates any sentences that use any character not in these lists. This both reduces any remaining cases of data in an incorrect language and incorrectly parsed markup from the web crawlers. It also reduces the effective character set remaining in the training data, which in turn reduces the effective vocabulary size of resultant MT systems, which we found to be beneficial when training modern NMT systems.

**Digit matching:** numbers, in particular digits, can be used to mark well matched sentences, and indeed they have been used as such in paired corpus alignment (Khadivi and Ney, 2005, Simard et. al 1992). Our system captures this by extracting all digits (in this case the characters 0-9) from the source and target sentence and eliminating them if they differ at all. This does introduce a small number of false positives where one side has the number in digits and the other in words ('1' vs 'one'), but we qualitatively found occurrences of this to be small.

### 2.2 Scoring Heuristics

To rank the remaining words, we turned to four heuristics we developed and found to be correlated with quality of the data.

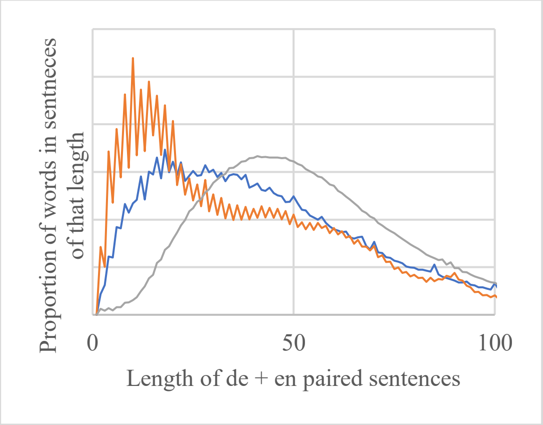


Figure 1: Plot of sentence length versus proportion of words that appear in sentences of that length, for the raw corpus (orange with leftmost peak), the corpus after our initial 'rules' (blue with central peak), and the dev data (grey with rightmost peak). With our sentence length heuristic we are trying to move the blue line to be closer to the grey one.

Each heuristic produced a score with a positive correlation to data quality (as measured by resultant BLEU), which we then scaled to be between 0 and 1. Our submissions were then based on weighted averages of those scores, where the weights between the different heuristics were determined empirically.

**Sentence length:** We noticed that the sentences in the corpus remaining after the rules were applied tended to be quite short. We confirmed this by comparing the sentence length distribution to that in the dev data (Figure 1). Note that our definition of sentence length here is the length of both source and target sentence summed, rather than length of one or the other.

These short sentences tended to be indicative of 'poor quality' and so we set up a heuristic to encourage longer sentences. In particular we use the following formula:

$$\begin{aligned}
&\mathit{if\ length} \leq 40: \\
&\quad \mathit{score} = \frac{2 * \mathrm{length}}{100} \\
&\mathit{elif\ length} \leq 80: \\
&\quad \mathit{score} = 0.8 * \frac{\mathrm{length} - 40}{200} \\
&\mathit{else}: \\
&\quad \mathit{score} = 1.0
\end{aligned} \tag{2}$$

We chose to use this relatively simple algorithm rather than any more sophisticated fitting technique in order to keep the system as general as possible. Any system which attempts to fit the exact curve is reliant on a target corpus which goes against the spirit of the task. We do note that we would probably not choose to use this heuristic in isolation however, as it would then essentially be no more than selecting the longest sentences.

**Perplexity:** perplexity measures have been used to filter language modelling corpora with respect to a specific domain (Gao et al, 2002; Lin et al., 1997). We would expect the same techniques to be beneficial here too. However, in the task description we were specifically asked not to use metrics related to domain-relatedness. As with our sentence length heuristic we look to mirror the overall perplexity statistics of a 'clean' corpus instead.

Rather than compare to a specific domain we trained a 5-gram using *KenLM* (Heafield et al, 2013) on the data itself, measured log(perplexity) of each sentence using this self-trained model and then did the same on the dev data. As with the sentence length heuristic, we found that the dev data displayed a slightly different behavior to the corpus being filtered (Figure 2) – in this case the overall shape of the graph was similar, peaked at a value of 0.82 for negative log perplexity divided by sentence length, but the dev data had a sharper peak, and the corpus to be filtered had more sentences of higher or lower perplexity values.

Our heuristic therefore upweights sentences closer to this peak, to try and match the dev data behavior.

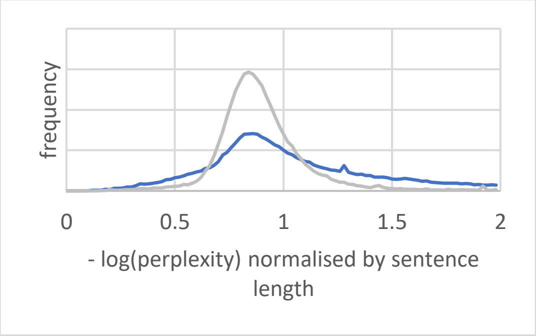


Figure 2: Plot showing frequency against negative log perplexity normalised by sentence length, for the corpus after rules were applied (blue, lower peak) and dev data (grey, higher peak). With our perplexity heuristic we are trying to move the blue line to be closer to the grey one.

$$\begin{aligned}&\mathit{if} - \log(\mathit{ppl})\ \mathit{per\ word} \leq 0.82:\\ &\quad \mathit{score} = 1 - \frac{(0.82 + \log(\mathit{ppl})\ \mathit{per\ word})}{0.82}\\ &\mathit{else}:\\ &\mathit{score} = \max(0,1 - \frac{-\log(\mathit{ppl})\mathit{per\ word} - 0.82}{3})\ (3)\end{aligned}$$

**Diversity:** following Song et. al (2014) we used sentence similarity in a rolling buffer to measure how diverse a sentence was compared to its neighbours.

Like Song et. al. we used a rolling window of 200 sentences, however we found that using BLEU to measure sentence similarity was too slow for practical use with such a large corpus. Instead we took a two-step approach, first checking if at least half of the words in the two sentences were in common. If so we then used simple edit distance to measure how similar the sentences were. The per sentence score derived from this heuristic was the minimum Levenshtein edit distance between a given sentence and all other sentences in its 200-sentence window.

To give this metric more chance of identifying similar sentences, we first sorted the entire corpus by sentence length, as sentences of similar length are more likely to have smaller edit distances.

This heuristic then effectively assigns high scores to sentences that exhibit distinctness to others in the corpus, whilst giving low scores to sentences that are near duplicates and hence adding little new information.

**MT filtering:** previous work has shown that machine translation systems themselves can directly be used to filter parallel corpora, either as a preprocessing step (Gaspers et. al. 2018) or even on the fly as part of the training process (Zhang et. al., 2017).

We therefore train an MT system on the entirety of the post-rules corpus. We then compute the one best translation for each sentence. Finally, we compute the decoder cost of both the one best translation and the reference translation. The decoder cost in this case is the cross-entropy loss. We did not normalize by sentence length as we found it made little difference.

The raw decoder cost of the reference translation by itself is an initially interesting metric, as low values correspond to sentences that are more likely to be correct translations as they don't diverge from what the system would expect to see. However, we also find that this approach biases the results towards short sentences that are very similar to one another, meaning resulting corpora lack diversity and fall foul of the rare words problem (Luong et. al 2015). The decoder cost of the 1-best translation is therefore used as a constraint on this. Our final score for this heuristic is the decoder cost of the reference sentence *minus* the decoder cost of the 1-best. We then compute this number in both translation directions and average.

High values of this derived score represent situations where the reference translation is judged much less likely than the 1-best by the decoder and thus should be discarded as likely junk. Very low scores show that the reference translation agrees with the model and are therefore unlikely to be junk. And further than that scores where the target has a lower cost than the target indicate explicit areas where the model needs to be improved – in other words exactly the sorts of inputs that are most valuable for the task of training a machine translation system.

We used the tensor2tensor framework to train a machine translation system for this scoring (Vaswani et al. 2018). The setup was the same as we used for benchmarking, as described in Section 3.

## 3 Benchmarking

To benchmark our progress, we use the tensor2tensor system (Vaswani et. al. 2018) which reports world leading results on machine translation tasks at present. We took the most recent commit of the code (at the time) from https://github.com/tensorflow/tensor2tensor/commit/99750c4b and used it without alteration.

We use this system without attempting to tune hyperparameters, except that we use the predefined 'transformer_small' recipe from the code repository (rather than the default 'transformer_base'), for speed and memory reasons. The 'transformer_small' recipe uses two hidden layers, each of size 256 and 4 attention heads. We trained each system for 500k steps (we found training for more steps was not helpful for performance) then averaged the last 8 checkpoints.

All BLEU scores reported used the described filtering system to prepare the training data, and then benchmark a trained transformer_small against the 'newstest2016' test set. BLEU was calculated using the t2t-bleu function in

|  | 1bn word corpus | dev corpus |
|---|---|---|
| Line length | 12.3% | 6.0% |
| Non-translation | 8.3% | 0.6% |
| Language identification | 12.0% | 1.5% |
| Character filtering | 24.9% | 13.5% |
| Digit matching | 26.5% | 6.3% |

Table 2: Percentages of the 1 billion word and dev corpora removed by each of the initial filtering

tensor2tensor and all reported numbers are on uncased text, with no tokenization applied.

## 4 Results

### 4.1 Initial 'rules'

Using the initial 'rules' removed 840 million words from the 1-billion-word corpus, leaving 160 million words for further scoring. Table 1 shows the contribution each rule made to this. Note that by contrast the rules would have removed a much smaller, but still significant, proportion of the dev data. This shows both that the rules are effective at removing 'bad' data (as we assume the 1 billion words contains more 'bad' data than the dev set) and that they are perhaps over aggressive and could bear some more tuning.

The rules were applied sequentially, so the latter rules may have removed more words if applied directly to the initial corpus.

Purely using these initial hard rules and then randomly selecting from the resulting 160M improves BLEU scores vastly compared to randomly selecting from the entire 1Bn word corpus (Table 2). For a target corpus size of 10M words the BLEU score improves from 5.93 to 26.14.

### 4.2 Scoring Heuristics

We applied the scoring heuristics described above in various combinations on the 160M words remaining after our initial 'rules'.

When filtering down to 100M words of data, any of the heuristics by themselves improved the BLEU score by between 0.12-1.67 as compared to randomly selecting from the 160M words post-'rules' (Table 2). Combining them in any combination gives further improvements and using all of them together gives a total of 1.97 gain in BLEU.

When filtering down to 10M words the picture is more complicated. Two of the heuristics by themselves produce worse BLEU scores (sentence length and MT scoring) and two improve the BLEU scores (perplexity and diversity). When combined equally there is a gain of 5.22, which is degraded if any of the metrics are omitted from that averaging. In particular the BLEU is degraded significantly if MT scoring is omitted from the combination.

We suspect that the very low scores exhibited in the 10M results are more than likely due to

| Method of filtering data down to target amount | 100M words | 10M words |
|---|---|---|
| Randomly selected sentences from initial 1Bn | * | 5.93 |
| Randomly selected sentences from 160M after initial 'rules' | 31.14 | 26.14 |
| Sentence length scoring used to pick best from 160M after 'rules' | 32.52 | 17.72 |
| Perplexity scoring used… | 32.81 | 29.00 |
| Diversity scoring used… | 31.80 | 28.46 |
| MT scoring used … | 32.26 | 17.07 |
| All four heuristics except length used… | 32.98 | 30.47 |
| All four heuristics except perplexity used… | 32.69 | 30.97 |
| All four heuristics except diversity used… | 32.71 | 30.34 |
| All four heuristics except MT used… | 32.83 | 17.86 |
| All four scoring heuristics averaged and used… | 33.11 | 31.36 |

Table 1: BLEU scores computed by training a tensor2tensor transformer_small system on 10M and 100M samples of data and then testing on newstest2016. The cell marked '*' could not be computed due to memory issues with our training setup. We list columns in terms of number of words in the corpus rather than the (perhaps more familiar) number of sentence pairs, as the task demanded we filter to a specific number of words rather than sentence pairs. The number of sentence pairs varied in each cell as different filtering techniques led to different average sentence lengths. The 10M corpora varied between 200k and 1M sentence pairs, for example, and the 100M corpora between 4M and 10M sentence pairs.

pathological failures in training the tensor2tensor system, however we were unable to ascertain the exact cause and found the numbers were reproduceable on multiple runs of training with the same setup and data.

### 4.3 Discussion

It is clear that using our initial 'rules' offer a significant improvement over random selection and that the scoring heuristics we have used are all capable of adding additional value in sub selecting data.

The sentence length scoring heuristic and the initial rules (barring language identification) are by some order of magnitude the fastest and simplest part of the system. For an initial look at data we would recommend using these before investing time into the more compute intensive rules.

Our entries to the competition were based on the balanced scoring across all four heuristics ('speechmatics-best-candidate-balanced-scoring.txt'), scoring purely based on the MT scoring ('speechmatics-purely-neural-scoring.txt') and a version with asymmetric weights heavily skewed towards the MT scoring ('speechmatics-prime-neural-scoring.txt').

### 4.4 Further Work

At present we have not tuned many parameters in our system. For optimal results we would spend more time on each of the 9 separate components we used for our system to optimize their various parameters with respect to final system BLEU.

For realistic use cases we would also expect that domain specific entropy filtering would be hugely beneficial, as we have previously found in language modelling (Williams et. al. 2015).

Conceptually we believe that the MT scoring heuristic has the most scope for future development. It is also the component most closely related to the actual task – translating text. Particularly interesting would be investigating its efficacy as the model capacity is scaled. Our belief is that some form of system that dynamically eliminates text as part of training could end up being the optimal approach to filtering out noisy parallel data.

## 5 Conclusion

The Speechmatics entry to the parallel corpus filtering task comprises a two-step system. The first step applies some simple rules to remove the bulk of the poor-quality data from a corpus. This gives most of the gains in terms of BLEU on a final trained system. We then apply four heuristics for scoring that give additional BLEU improvements.

We believe this is a relatively straightforward system that can be used across a wide variety of language pairs with little alteration to produce high quality reduced size MT corpora.

## References

Michael Denkowski, Gred Hanneman, Alon Lavie. (2012). The CMU-Avenue French-English Translation System. *Proceedings of the NAACL 2012 Workshop on Statistical Machine Translation*, Montreal, Canada.

Jianfend Gao, Joshua Goodman, Mingjing Li and Kai-Fu Lee. (2002). Toward a unified approach to statistical language modeling for Chinese. *ACM Transactions on Asian Language Information Processing (TALIP)*. ACM, New York, USA

Judith Gaspers, Penny Karanasou, Rajen Chetterjee. (2018) Selecting Machine-Translated Data for Quick Bootstrapping of a Natural Language Understanding System. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. New Orleans, USA

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan Clark and Philipp Koehn. (2013). Scalable Modified {Kneser-Ney} Language Model Estimation. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria.

Armand Joulin, Edouard Grave, Piotr Bojanowski and Tomas Mikolov. (2017). Bag of Tricks for Efficient Text Classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain

Shahram Khadivi and Hermann Ney. (2005). Automatic Filtering of Bilingual Corpora for Statistical Machine Translation. *In: Montoyo A., Muńoz R., Métais E. (eds) Natural Language Processing and Information Systems. NLDB 2005. Lecture Notes in Computer Science, vol 3513*. Springer, Berlin, Heidelberg

Sung-Chien Lin, Chi-Lung Tsai, Lee-Feng Chien, Keh-Jiann Chen, Lin-Shan Lee. (1997). Chinese language model adaptation based on document classification and multiple domain-specific language models. *Proceedings of the 5th European*

*Conference on Speech Communication and Technology*. Rhode, Greece.

Marco Lui and Timothy Baldwin. (2012). langid.py: An Off-the-shelf Language Identification Tool. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012), Demo Session.* Jeju, Republic of Korea

Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals and Wojciech Zaremba. (2015). Addressing the Rare Word Problem in Neural Machine Translation. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Beijing, China

Michel Simard, George Foster, Pierre Isabelle. (1992). Using cognates to align sentences in bilingual corpora. *Fourth Int. Conf. on Theoretical and Methodological Issues in Machine Translation (TMI92).* Montreal, Canada.

Xingyi Song, Trevor Cohn and Lucia Specia. (2014). Data Selection for Discriminative Training in Statistical Machine Translation. *17th Annual Conference of the European Association for Machine Translation.* Dubrovnik, Croatia

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer and Jakob Uszkoreit. (2018) Tensor2Tensor for Neural Machine Translation. *CoRR, abs/ 1803.07416*.

Will Williams, Niranjani Prasad, David Mrva, Tom Ash, Tony Robinson. (2015) Scaling recurrent neural network language models. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Brisbane, Australia.

Dakun Zhang, Jungi Kim, Josep Crego, Jean Snellart. (2017) Boosting Neural Machine Translation. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Taipei, Taiwan.

# STACC, OOV Density and N-gram Saturation: Vicomtech's Participation in the WMT 2018 Shared Task on Parallel Corpus Filtering

**Andoni Azpeitia** and **Thierry Etchegoyhen** and **Eva Martínez Garcia**
Vicomtech
Mikeletegi Pasalekua, 57
Donostia / San Sebastián, Gipuzkoa, Spain
{tetchegoyhen, aazpeitia, emartinez}@vicomtech.org

## Abstract

We describe Vicomtech's participation in the WMT 2018 Shared Task on parallel corpus filtering. We aimed to evaluate a simple approach to the task, which can efficiently process large volumes of data and can be easily deployed for new datasets in different language pairs and domains. We based our approach on STACC, an efficient and portable method for parallel sentence identification in comparable corpora. To address the specifics of the corpus filtering task, which features significant volumes of noisy data, the core method was expanded with a penalty based on the amount of unknown words in sentence pairs. Additionally, we experimented with a complementary data saturation method based on source sentence n-grams, with the goal of demoting parallel sentence pairs that do not contribute significant amounts of yet unobserved n-grams. Our approach requires no prior training and is highly efficient on the type of large datasets featured in the corpus filtering task. We achieved competitive results with this simple and portable method, ranking in the top half among competing systems overall.

## 1 Introduction

Data-driven approaches to Machine Translation (MT) have been the dominant paradigm in the last two decades, with the development of Statistical Machine Translation (SMT) (Brown et al., 1990), and, more recently, of Neural Machine Translation (NMT) (Bahdanau et al., 2015). These approaches require large volumes of parallel sentences to properly model translation in a given language pair. However, large quality parallel corpora based on human translations are scarce across language pairs, and there is a strong need to build clean corpora from different sources.

The World Wide Web is a rich source of multilingual data, from which parallel corpora can be automatically created under appropriate conditions of use (Forcada et al., 2016). However, corpora created via crawling, with automated document and sentence alignment, tend to exhibit significant volumes of noisy data, which can be detrimental to the training of MT systems (Khadivi and Ney, 2005; Khayrallah and Koehn, 2018a).

The task of cleaning noisy data from parallel corpora has been tackled by various researchers over the years. In (Munteanu and Marcu, 2005), noise removal is performed via a maximum entropy model trained on observations of clean and noisy data. Esplá-Gomis and Forcada (2009) include sentence alignment scores in BiTextor, a tool that performs the complete chain of corpus creation from web data, to filter dubious sentence pairs. In (Khadivi and Ney, 2005), two approaches are evaluated, based on length and on lexical translation likelihood, showing statistically significant improvements in translation quality using the filtered corpus. An unsupervised filtering method based on outlier detection is proposed in (Taghipour et al., 2011), who also report improvements in translation quality from their filtered corpus. In (Cui et al., 2013), the approach to data filtering is based on graph-based random walks, with improvements observed for Chine-English machine translation. Recently, Xu and Koehn (2017) introduced Zipporah, a fast data selection system for noisy parallel corpora, which is shown to result in improved SMT system quality.

The WMT 2018 task on parallel corpus filtering offers the possibility to compare different approaches to the task, evaluating their impact on both SMT and NMT systems on several test sets in different domains. Our participation in the task aimed to evaluate a simple and portable approach, based on the efficient STACC system for parallel sentence extraction from comparable corpora (Etchegoyhen and Azpeitia, 2016). We extended

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 860–866
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64100

the original approach with a simple method based on the number of unknown words, to tackle the significant amounts of noise featured in the corpus filtering task. Additionally, we experimented with a simple approach to data redundancy, based on n-gram saturation. Our contribution centred on providing a sound method that can be easily deployed, does not require prior training, and can efficiently process large volumes of data.

## 2 Approach

Our approach to the task is based on STACC, a portable and efficient method for the identification of parallel sentences in comparable corpora (Etchegoyhen and Azpeitia, 2016) which obtained the best results for all language pairs in the BUCC shared tasks (Azpeitia et al., 2017, 2018). As the method assigns an alignment score to source and target sentence pairs, it can be directly applied to parallel corpus filtering as well, with a simple extension for this specific task. We describe the components of our approach in the next sub-sections.

### 2.1 STACC

The STACC approach has been described and explored in detail in (Etchegoyhen and Azpeitia, 2016), and we briefly summarise below how similarity is computed with this method.

Let $s_i$ and $s_j$ be two tokenised and truecased sentences in languages $l_1$ and $l_2$, respectively, $S_i$ the set of tokens in $s_i$, $S_j$ the set of tokens in $s_j$, $T_{ij}$ the set of lexical translations into $l_2$ for all tokens in $S_i$, and $T_{ji}$ the set of lexical translations into $l_1$ for all tokens in $S_j$.[1]

Lexical translations are initially computed from sentences $s_i$ and $s_j$ by retaining the $k$-best translations for each word, if any, as determined by the ranking obtained from the lexical translation probabilities computed with IBM word alignment models (Brown et al., 1990). The sets $T_{ij}$ and $T_{ji}$ that comprise these $k$-best lexical translations are then expanded by means of two operations:

1. For each element $x$ in the set difference $T'_{ij} = T_{ij} - S_j$ (respectively $T'_{ji} = T_{ji} - S_i$), and each element $y$ in $S_j$ (respectively $S_i$), if $x$ and $y$ share a common prefix of more than $n$ characters, the prefix is added to both $T_{ij}$ and $S_j$ (respectively $T_{ji}$ and $S_i$). This longest common prefix matching strategy is meant to capture morphological variation via minimal computation.

2. Numbers and capitalised truecased tokens not found in the translation tables are added to the expanded translation sets $T_{ij}$ and $T_{ji}$. This operation addresses named entities, which are strong indicators of potential alignment given their low relative frequency and are likely to be missing from translation tables trained on different domains.

With source and target sets as defined here, the STACC similarity score is then computed as in Equation 1:

$$stacc(s_i, s_j) = \frac{\frac{|T_{ij} \cap S_j|}{|T_{ij} \cup S_j|} + \frac{|T_{ji} \cap S_i|}{|T_{ji} \cup S_i|}}{2} \qquad (1)$$

Similarity for the core metric is thus defined as the average of the Jaccard similarity coefficients obtained between sentence token sets and expanded lexical translations in both directions.

The STACC approach has been extended in (Azpeitia et al., 2017, 2018), notably via a word weighting scheme that led to significant improvements in the parallel sentence extraction task. In this work, we used the original weightless approach, as it performed slightly better in preliminary experiments on the noisy web data of the WMT 2018 task.

### 2.2 OOV Density

The corpus for the WMT 2018 shared task on parallel corpus filtering features significant volumes of noise, as is typical with parallel corpora gathered via web crawling that targets recall. (Khayrallah and Koehn, 2018b) manually examined a sample of data generated by the Paracrawl project,[2] of the type used in this shared task, and identified as noise misaligned sentences, content in the wrong languages, untranslated sentences, random byte or HTML markup sequences. The latter four types can be notably characterised as displaying significant percentages of out-of-vocabulary (OOV) words, assuming a vocabulary extracted from a separate parallel corpus with limited amounts of noisy data.

As previously described, the STACC approach, which constitutes the core of our method, is geared

[1] As in the original approach, we use sets rather than multisets, i.e. without repeated elements. The term *tokens* refers to the components of the tokenised sentences, and repeated tokens are thus only represented once in the sets.

[2] https://paracrawl.eu/

towards computing alignment scores in comparable corpora, with lower volumes of noise, notably allowing OOV words to contribute to the score if they are capitalised words in truecased sentences or numbers. This enables the capture of surface-defined named entities, which are a decisive factor for parallel sentence identification in comparable datasets (Azpeitia et al., 2018). However, this approach can be weaker in highly noisy datasets, where, for instance, random sequences of numbers may lead to an unwarranted high alignment score.

Since we aimed to avoid adding task-specific cleanup heuristics, such as performing time-consuming language identification or filtering sequences in an ad-hoc manner, we experimented with a penalty based on the number of unknown words in the corpus to be filtered, determined from the separate parallel corpus used to extract lexical translations. The penalty is computed as follows for each sentence $s$, source or target, where $|oov|$ is the number of unknown words in the sentence and $|s|$ is the sentence length, in number of words:

$$p(s) = 1 - \frac{|oov|}{|s|} \tag{2}$$

The STACC.OOV alignment score for each sentence pair ($s_i$, $s_j$) is then computed as follows:

$$stacc.oov(s_i, s_j) = stacc(s_i, s_j) \cdot \frac{p(s_i) + p(s_j)}{2} \tag{3}$$

Thus, sentences with a small amount of OOV words, of interest to extend MT coverage, will be assigned a score close to the original STACC score, whereas the score for dubious sentences with large numbers of unknown words will tend to zero. Our primary submission was based on the metric in Equation 3, as the initial goal of the task was to assign an absolute alignment quality score.

### 2.3 N-gram Saturation

The organisers of the shared task had allowed the use of metrics that did not score sentences in isolation. That is, sentence pairs could be scored by considering their redundancy with regards to higher scoring pairs. This aspect enables the design of methods that select the n-best sentence pairs to train machine translation models.

To experiment with data redundancy, we implemented a simple method based on n-gram coverage, similar in spirit to the n-gram coverage and saturation methods of Eck et al. (2005) and Lewis and Eetemadi (2013). The method can also be related to the Feature Decay approach proposed in (Biçici and Yuret, 2011), originally applied to SMT models and recently evaluated on NMT as well (Poncelas et al., 2018).

We compute n-gram saturation by first sorting the corpus according to the STACC.OOV scores, from high to low scores. We then process the sorted corpus by extracting n-grams (up to a specific order $n$) from each source sentence, storing the collected n-grams in a Patricia trie $T$ (Morrison, 1968) for fast retrieval, and computing the amount of new n-grams for each sentence. The steps for a given sentence $s$ are described below:

1. Retrieve all n-grams in $s$.

2. Determine all new n-grams from step 1, i.e. n-grams not found in the trie $T$.

3. Compute the ratio of new to existing n-grams in $s$ as in Equation 4, for each n-gram $ng$ up to order $k$:

$$ngsat(s) = \frac{\sum_{n=1}^{k} ng_n \notin T}{\sum_{n=1}^{k} ng_n \in T} \tag{4}$$

4. Add all new n-grams to the trie $T$.

Finally, we compute the score of the STACC.OOV.NGSAT variant for each sentence pair by multiplying the pair's existing score in the sorted corpus, computed as in Equation 3, by its *ngsat* score. Thus, pairs that provide no new n-grams would get an overall score of zero, while pairs with a large amount of new n-grams would get a score close to the existing score.

This simple method differs from the one in (Eck et al., 2005) in two ways: we do not pre-compute nor use n-gram frequency, and our normalisation factor is the total number of n-grams for the sentence instead of sentence length. Our approach also has linear complexity instead of quadratic, since, contrary to their different scenario focussed on data selection, we do not need to recalculate costs for all sentence pairs after processing one pair. Our method also differs from that of (Lewis and Eetemadi, 2013), as we do not use a threshold of n-gram counts but the percentage of new n-grams contributed by a given sentence,

| MT | SYSTEM | AVG | RANK | NEWS | IWSLT | ACQUIS | EMEA | GLOBAL | KDE |
|---|---|---|---|---|---|---|---|---|---|
| SMT 10M | BEST | 24.58 | 1/48 | 29.59 | 22.16 | 21.45 | 28.28 | 22.67 | 25.51 |
| SMT 10M | STACC.OOV | 23.25 | 16/48 | 27,48 | 20.42 | 19.33 | 26.51 | 21.20 | 24.55 |
| SMT 10M | STACC.OOV.NGSAT | 23.29 | 13/48 | 27,52 | 19.80 | 19.33 | 26.84 | 21.12 | 25.14 |
| SMT 100M | BEST | 26.50 | 1/48 | 31.35 | 23.17 | 22.51 | 31.45 | 24.00 | 26.93 |
| SMT 100M | STACC.OOV | 25.91 | 24/48 | 30.47 | 22.47 | 22.16 | 30.30 | 23.43 | 26.63 |
| SMT 100M | STACC.OOV.NGSAT | 25.80 | 29/48 | 30.17 | 22.39 | 22.12 | 30.03 | 23.36 | 26.70 |
| NMT 10M | BEST | 28.62 | 1/48 | 36.04 | 25.23 | 25.30 | 32.72 | 26.72 | 28.25 |
| NMT 10M | STACC.OOV | 26.35 | 13/48 | 32.33 | 22.57 | 22.55 | 28.96 | 24.28 | 27.39 |
| NMT 10M | STACC.OOV.NGSAT | 25.64 | 17/48 | 31.25 | 21.81 | 20.67 | 29.09 | 23.48 | 27.56 |
| NMT 100M | BEST | 32.06 | 1/48 | 39.85 | 27.43 | 28.36 | 36.70 | 29.26 | 30.79 |
| NMT 100M | STACC.OOV | 30.40 | 27/48 | 37.08 | 26.35 | 26.81 | 34.54 | 27.74 | 29.89 |
| NMT 100M | STACC.OOV.NGSAT | 24.91 | 40/48 | 27.23 | 22.44 | 23.15 | 26.92 | 22.94 | 26.76 |

Table 1: Results on the WMT 2018 test sets

| MT | SYSTEM | $\Delta_{\text{MEAN}}$ | $\Delta_{\text{MEDIAN}}$ | $\Delta_{\text{BEST}}$ |
|---|---|---|---|---|
| SMT 10M | STACC.OOV | +1.83 | +0.74 | -1.33 |
| SMT 10M | STACC.OOV.NGSAT | +1.87 | +0.79 | -1.29 |
| SMT 100M | STACC.OOV | +1.03 | +0.03 | -0.59 |
| SMT 100M | STACC.OOV.NGSAT | +0.92 | -0.08 | -0.71 |
| NMT 10M | STACC.OOV | +4.51 | +1.79 | -2.27 |
| NMT 10M | STACC.OOV.NGSAT | +3.80 | +1.09 | -2.98 |
| NMT 100M | STACC.OOV | +2.47 | -0.27 | -1.65 |
| NMT 100M | STACC.OOV.NGSAT | -3.03 | -5.77 | -7.15 |
| ALL | STACC.OOV | +2.46 | +0.57 | -1.46 |
| ALL | STACC.OOV.NGSAT | +0.89 | -0.99 | -3.03 |

Table 2: Scoring differences on core statistics

and also assume the initial ordering provided by the STACC.OOV scores. Finally, our approach differs from the Feature Decay method in (Biçici and Yuret, 2011) on several aspects, as it is not based on rate of decay and n-gram saturation scores are computed in a single pass on the corpus to be filtered, without referring to source test features.

Our goal in experimenting with n-gram saturation was mainly to include a low complexity method that could account for data redundancy in a simple way. The scope of the experiments was also reduced to only cover n-grams on the source side, as we meant to evaluate the impact of data redundancy in terms of source context coverage. This evidently excludes cases where a saturated source context can be translated differently in the target language, which can impact the number of learned translation options and subsequently affect evaluation scores. We leave further evaluations of such cases for future research. In the next sections, we evaluate the STACC.OOV.NGSAT variant as our secondary submission to the WMT 2018 task.

## 3 Experimental Setup

Our approach implies only minimal deployment settings. We ran STACC with the following two hyper-parameters: minimal prefix length was set to $4$ and $k$-best translation lists limited to $5$ candidates. For the STACC.OOV.NGSAT variant, the n-gram order was set to 3.

For the lexical translation tables needed by the STACC algorithm, we trained IBM2 models with the FASTALIGN toolkit (Dyer et al., 2013), on corpora made available for the WMT 2018 news translation task. The corpora thus included *Europarl v7*, *Common Crawl*, *NewsCommentary*, and the *Rapid corpus of EU press releases*. The *Paracrawl* corpus was excluded from the training data in order to extract reliable lexical translation tables from less noisy bilingual corpora. After duplicates removal, the training corpus amounted to $5,623,721$ parallel sentences.

The corpus was processed on an in-house server, using 64 threads. The total processing time for the 104 million sentence pairs of the corpus was around 57 minutes with the STACC.OOV variant, consuming a maximum of 11.3GB of RAM. With the STACC.OOV.NGSAT variant, processing time was approximately 5 times slower, with an order of magnitude larger consumption of RAM, mainly due to our online trie computation.

Given our stated objectives of evaluating a simple and portable method for the task, our preliminary experiments were all based on variants of the STACC approach, evaluated on the development set provided by the organisers. We notably experimented with the variant in (Azpeitia et al., 2017), where the STACC score is computed via frequency-based lexical weighting that favours content words, and the variant in (Azpeitia et al., 2018), which features a scoring penalty that promotes named-entity matching. Although the differences were minor, the original STACC approach performed better overall and was thus selected as the core of the metric for our final submissions.

## 4 Results

The results of our approach on the WMT 2018 test sets are shown in Table 1.[3] Overall, our primary submission, STACC.OOV performed well on the task, ranking in the top third for SMT 10M and NMT 10M, and as a mid-performing system in the other two scenarios. Given the simplicity and efficiency of our approach, and the relatively minor differences with the top performing systems, we view these results as quite satisfactory.

The ranking was relatively uniform between test sets, with the notable exception of the KDE test set for which our approach was among the top 10 submissions in 3 out of 4 scenarios, and ranked 20th in the fourth case. This may be due to the fact that our scores are assigned purely in terms of alignment and not geared towards selecting sentence pairs that may be more informative for the news domain or similar, for instance. Thus, short sentence pairs with technical content that are correct translations will receive high scores although they may not be the most relevant pairs for the other test sets that feature less technical language.

Both systems performed similarly for SMT and NMT in terms of rankings obtained on the 10M and 100M versions. The variant of our approach that includes n-gram saturation performed similarly to our primary submission overall for SMT, but worse for NMT. Given these results and the fact that computing n-gram saturation is more resource-consuming, our primary submission was the optimal option of the two. A more detailed analysis would be needed to evaluate the causes for the drop caused by n-gram saturation for NMT. It could be conjectured that NMT training is optimal with the largest number of contextual variants in the training data, variants which would tend to be demoted via n-gram saturation. Phrase-based SMT can be considered less sensitive to contextual variants, given its core phrase-independence translation assumption. We leave a more precise analysis of these aspects for future research.[4]

To further compare our systems to the other submissions, we computed the core statistics on the average scores of all systems. In Table 2 we indicate the differences between our submission scores and the mean ($\Delta_{\text{MEAN}}$), median ($\Delta_{\text{MEDIAN}}$) and best ($\Delta_{\text{BEST}}$) scores. In the last two rows of the table, we indicate the average differences for all scenarios in each category.

Our system performed better than the mean, in particular for NMT, with improvements of 4.51 and 2.47 for the primary submission. The one exception is the n-gram saturation variant, whose performance dropped significantly for NMT 100M, which may be explained under the aforementioned conjecture. The results in terms of the median are in line with the rather similar results obtained by a large number of participating systems.

Another notable aspect illustrated by this view of the results is the relatively higher differences with respect to the best performing system when considering NMT results, with a 1 BLEU point difference on average. Determining whether this difference reflects a systematic tendency would require a larger set of experiments with different corpora and language pairs. On average, our primary submission was 1.46 BLEU points below the best system and 2.46 points above the mean. Considering also the high efficiency of the approach, which

[3] For ease of presentation, we only indicate the official results in terms of C-BLEU scores, as provided by the shared task organisers. Along with the results of our systems, we also indicate the scores of the best system for each test set. The column AVG indicates the average score across all test sets and RANK denotes the ranking of the system among all participants according to the average score.

[4] As pointed out by one of the reviewers, an alternative explanation could be formulated. SMT systems are more sensitive to missing n-grams, contrary to NMT models, which rely on word embeddings and are thus less sensitive to specific words or n-grams. Thus, rather than NMT needing more contextual variants, the results could reflect that SMT benefits more from the additional n-grams provided via the saturation method, whereas NMT suffers from the imbalance in the training data that results from n-gram saturation filtering. Although this explanation has its merits, it would also warrant further examination. First, our results did not actually improve when using n-gram saturation for SMT, overall, which would tend to show that the additional n-grams collected via saturation did not have a significant impact in these experiments; only NMT systems were negatively impacted by the saturation method. Secondly, the suggested data imbalance could actually be viewed as a reduction in contextual variants, as we hypothesised, which could impact the computation of both embeddings and context vectors in attention-based NMT. Whether data imbalance could be viewed differently from contextual variants reduction is an interesting topic to be further explored.

can process the 104M parallel sentences in under an hour without the need for language-dependent tools nor any prior training, we view our method as a practical and reliable alternative to filter large noisy parallel corpora.

## 5 Conclusions

We have described our participation in the WMT 2018 shared task on parallel corpus filtering. Our approach was based on the STACC system, which only requires lexical translation tables to assign alignment quality scores. For this task, the core system was augmented with a simple penalty based on the number of unknown words in the sentences, to account for the significant volumes of noise in the corpus. Additionally, we experimented with a simple n-gram saturation scheme to evaluate the impact of demoting redundant data.

The results were satisfactory for such a simple and computationally efficient approach, which does not require prior training, sophisticated setups, language-dependent analysers, complex feature sets or extensive computational resources. In fact, our approach only requires pre-trained IBM2 lexical translation tables, which can be efficiently computed with generic off-the-shelf tools. We achieved competitive results overall, ranking in the top half among competing systems overall, with scores above the mean and less than 1.5 BLEU points below the top performing systems on average. The n-gram saturation variant did not provide significant improvements and actually performed significantly worse in one scenario, while also consuming more computational resources. The simpler primary variant of the system thus proved optimal for the task and more research would be needed to better account for data redundancy within our core approach.

The system we submitted is also quite efficient, being able to process the 104M sentence pairs in the task corpus in under an hour. Overall, we view our approach as a portable and efficient method to filter noisy data from parallel corpora. In future work, we will evaluate variants of the approach, exploring in particular the specifics of the data featured in different domains.

**Acknowledgements** We wish to thank the anonymous WMT 2018 reviewers for their detailed and helpful reviews of this work. Opinions, errors and omissions are our own.

## References

Andoni Azpeitia, Thierry Etchegoyhen, and Eva Martínez Garcia. 2017. Weighted Set-Theoretic Alignment of Comparable Sentences. In *Proceedings of the Tenth Workshop on Building and Using Comparable Corpora*, pages 41–45.

Andoni Azpeitia, Thierry Etchegoyhen, and Eva Martínez Garcia. 2018. Extracting Parallel Sentences from Comparable Corpora with STACC Variants. In *Proceedings of the Eleventh Workshop on Building and Using Comparable Corpora*, pages 48–52.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Ergun Biçici and Deniz Yuret. 2011. Instance Selection for Machine Translation using Feature Decay Algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283.

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A Statistical Approach to Machine Translation. *Computational linguistics*, 16(2):79–85.

Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li, and Ming Zhou. 2013. Bilingual data cleaning for SMT using graph-based random walk. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 340–345.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Low cost portability for statistical machine translation based on n-gram coverage. In *Proceedings of MTSummit X*.

Miquel Esplá-Gomis and Mikel L Forcada. 2009. Bitextor, a free/open-source software to harvest translation memories from multilingual websites. *Proceedings of MT Summit XII*.

Thierry Etchegoyhen and Andoni Azpeitia. 2016. Set-Theoretic Alignment for Comparable Corpora. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1: Long Papers, pages 2009–2018.

Mikel L Forcada, Miquel Espla-Gomis, and Juan Antonio Perez-Ortiz. 2016. Stand-off annotation of web content as a legally safer alternative to bitext crawling for distribution. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 152–164.

Shahram Khadivi and Hermann Ney. 2005. Automatic filtering of bilingual corpora for statistical machine translation. In *International Conference on Application of Natural Language to Information Systems*, pages 263–274. Springer.

Huda Khayrallah and Philipp Koehn. 2018a. On the impact of various types of noise on neural machine translation. *CoRR*, abs/1805.12282.

Huda Khayrallah and Philipp Koehn. 2018b. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*.

William Lewis and Sauleh Eetemadi. 2013. Dramatically reducing training data size through vocabulary saturation. In *Proceedings of the Eight Workshop on Statistical Machine Translation*, pages 281–291.

Donald R Morrison. 1968. Patricia practical algorithm to retrieve information coded in alphanumeric. *Journal of the ACM (JACM)*, 15(4):514–534.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. 2018. Feature decay algorithms for neural machine translation.

Kaveh Taghipour, Shahram Khadivi, and Jia Xu. 2011. Parallel corpus refinement as an outlier detection algorithm. *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 414–421.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950.

# A hybrid pipeline of rules and machine learning to filter web-crawled parallel corpora

**Eduard Barbu**
Institute of Computer Science
University of Tartu
Tartu, Estonia
eduard.barbu@ut.ee

**Verginica Barbu Mititelu**
Research Institute for Artificial Intelligence
Romanian Academy
Bucharest, Romania
vergi@racai.ro

## Abstract

A hybrid pipeline comprising rules and machine learning is used to filter a noisy web English-German parallel corpus for the Parallel Corpus Filtering task. The core of the pipeline is a module based on the logistic regression algorithm that returns the probability that a translation unit is accepted. The training set for the logistic regression is created by automatic annotation. The quality of the automatic annotation is estimated by manually labeling the training set.

## 1 Introduction

The task "Parallel Corpus Filtering" presented a noisy web crawled parallel corpora (English-German) whose English side contains one billion words. The participants had to select two "clean" subsets consisting of 10 million words and 100 million words, respectively. The quality of the two subsets was determined by the BLEU score of a statistical machine translation (based on Moses) and a neural machine translation system (Marian) trained on these subsets. The BLEU scores were computed for multiple not disclosed sets.

The parallel corpus filtering task bears similarity to translation memory cleaning task and Quality estimation task.

Some systems that spot false translation units in translation memories are surveyed in Barbu (2016). One of the most successful systems is trained not only on features related to translation quality, but also on features related to grammatical errors and features related to fluency and lexical choice (Wolff, 2016).

Given the similarity between the translation memory cleaning task and this task we have adapted part of our system for cleaning the translation memories. The system requires supervision and word alignment knowledge. However, the "Parallel Corpus Filtering" task specifications restrict the usage of external parallel corpora and allow minimum alignment information. Therefore, we had to re-engineer the above mentioned system and produce a pipeline that respects the task requirements.

In the next section we present the re-engineered pipeline. The section 3 shows an in-house evaluation and in the last section we draw the conclusions.

## 2 Pipeline description

The pipeline for finding the best translation units for the Parallel corpus filtering task is shown in figure 1. The pipeline consists of three modules, which we describe below.

The module **Filtering Rules** filters those translation units that are not good to train machine translation systems on because they are either too short or are prone to errors. The discarded units have less than 10 words in source or target, or the language codes assigned by the language detector Cybozu[1] do not coincide with the expected language codes ("en" for the source segment and "de" for the target segment), or have a Church-Gale score (Gale and Church, 1993) that is less than $-4$ or greater than $4$. In the task submission the translation units that do not fulfill the above criteria have a score equal to $-100$. The initial number of translation units is 104.002.521. After filtering there remain 11.030.014 translation units. The English side of the remaining units contains 269.949.547 words corresponding to 241.984.520 German words. From this filtered set the two subsets required by the task description are selected.

The module **Machine Learning** is the core of the pipeline. Because the manual annotation was

[1] https://github.com/shuyo/language-detection

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 867–871
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64101

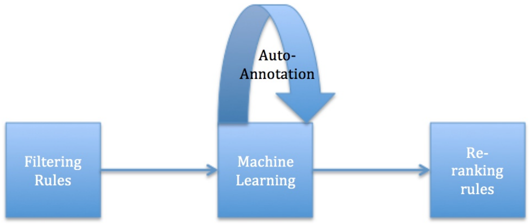


Figure 1: The pipeline for translation units selection

not allowed, the training set was generated by a simple heuristic rule. From the translation units not scored by the previous module we have drawn randomly approximately 1700 translation units. These units are annotated automatically in the following way. If the Hunalign (Varga et al., 2005) score, provided with the test file, for a translation unit is higher than a fixed threshold 0.9[2] we consider that translation unit as a positive example. If the Hunalign score is less or equal to the threshold the translation unit is a negative example. In section 3 we evaluate how accurate the automatic annotation is.

The **Machine Learning** module uses three kinds of features: *Presence/Absence* features, *Alignment Features* and *Fluency Features*. The feature values are all numerical because for classification we use scikit-learn machine learning toolkit (Pedregosa et al., 2011).

1. **Presence/Absence** features. This category of features signal the presence/absence of an entity in source or target segments. The features capture the intuition that if an entity is present in the source segment and if the target segment is a translation of the source segment it is very probable that the same entity is present in the target segment.

   - *Entity Features*. These features are *tag*, *URL*, *email*, *name entity*, *punctuation*, *number*, *capital letters*, *words in capital letters*. The value of these features is 1 if the source or target segments contain a tag, URL, email, name entity, punctuation, capital letters or words written in capital letters, otherwise is 0.
   - *Entity Similarity Features*. For features *tag*, *URL*, *email*, *name entity*, *punctuation*, *number*, the cosine similarity between the source and target segments entity features vectors is computed. If the respective features are present in the source segment and the target segment is the translation of the source we expect that the system learns the range of the admissible similarity values.
   - *Capital letters word difference*. The value of this feature is the ratio between the difference of the number of words containing at least a capital letter in the source segment and the target segment and the sum of the capital letter words in the translation unit. It is complementary to the feature *capital letters*.
   - *Only capital letters difference*. The value of the feature is the ratio between the difference of the number of words containing only capital letters in the source segment and the target segments and the sum of only the capital letter words in the translation unit. It is complementary to the feature *words in capital letters*.

2. **Alignment Features**. The idea behind alignment features is that sentence alignments, or the information that can help to decide if an alignment is likely or not, provide an important clue for the hypothesis that source and target segments are translations.

   - *language difference*. If the language codes identified by Cybozu language detector for the source and target segments coincide with the language codes declared for the same source and target segments, then the feature value is 1, otherwise the feature value is 0. As we have seen, the English and German segments have more than 10 words, therefore the language detector has enough information to return the segment language with good precision.
   - *Gale-Church score*. This feature is the slightly modified Gale-Church score described in the equation 1 and introduced in (2011). This score reflects the idea that the length of the source ($l_s$) and target segments ($l_d$) that are true translations is correlated. We expect that the classifiers learn the threshold that

[2] The threshold value comes from our previous experience with Hunalign aligner.

separates the positive and negative examples. However, relying exclusively on the Gale-Church score is tricky because there are cases when a high Gale-Church score is perfectly legitimate. For example, when the acronyms in the source language are expanded in the target language.

$$CG = \frac{l_s - l_d}{\sqrt{3.4(l_s + l_d)}} \qquad (1)$$

- *Hunalign score*. This is the score returned by Hunalign sentence aligner and was provided by the task organizers. The score depends on the quality of the English-German dictionary used by the aligner.

3. **Fluency Features**. These features values correlate with fluency of the translation units in source and target languages.

   - *Perplexity* To capture the fluency of the source and target we compute the perplexity of the segments in English and German using KenLM toolkit (Heafield, 2011). The KenLM language model was trained as advised on the shared-task web page - on the WMT 2018 news translation task data for German-English from which we have eliminated the Paracrawl parallel corpus. Moreover, we have also run Cybozu language detector to eliminate sentences that are not identified as written in English or German. Thus, the English corpus for training KenLM language model has 5.802.775 sentences and 126.831.658 words and the German corpus has 5.673.375 and 116.360.460 words.

The classification algorithm used by the **Machine Learning** module is logistic regression. For each filtered translation unit this module outputs the probability score that the respective unit is positive. One hopes that this probability score correlates with the translation unit quality.

The last module, **Re-ranking rules**, comprises a set of rules to re-score the probability scores outputted by the previous module. It implements the following rules :

1. *Same Digits Rule*. This rule states that if the target segment is a translation of the source segment, and the source segment contains some digits, then the target segment should contain the same digits, possibly in a different order. If this is not the case, the translation unit is re-scored by subtracting 1 from its probability score. Please, notice that the rule allows for the dates to be written in different formats. For example, if the source segment contains the date “02/01/2001” (format mm/dd/yyyy) and in the target segment the date is written as “01/02/2001” (format dd/mm/yyyy), then the translation unit is not re-scored.

2. *Same Numbers*. This rule states that if the target segment is a translation of the source segment, and the source segment contains some numbers, then the target segment should contain the same numbers possibly in a different order. If a translation unit passes the first rule by chance and if it does not pass this rule, then it will be downgraded subtracting 1 from is probability score.

3. *Rule URL*. This rule applies to those translation units that contain Uniform Resource Locators like web addresses, for example. If the length of the web address is longer than the portion of normal text in the source or target segment, then the translation unit is re-scored by subtracting 1 from its probability score.

4. *Rule Tags*. If the source and target segments are translations and they contain tags, then we expect that the tags are the same. If this is not the case 1 is subtracted from the translation unit probability score.

Finally, to ensure diversity among the best rated translation units that comprise the first evaluated set containing 10 millions of words we compute the cosine similarity between the English segments. We keep in the first set only those translation units whose cosine similarity (computed between English segments) is less than 0.85[3]

[3] To compute the cosine matrix we have used “TfidfVectorizer” from “sklearn”. Unfortunately, on our server we could not compute the matrix for all units and had to restrict to compute matrices with 30000 lines.

| Confusion Matrix | Predicted 1 | Predicted 0 |
|---|---|---|
| Actual 1 | 1010 | 233 |
| Actual 0 | 67 | 404 |

Table 1: The Confusion Matrix

## 3 Evaluation

We have manually annotated the automatically annotated pairs used to train the logistic regression algorithm. A non-native German language speaker has annotated this set with the label ”1” if the translation unit is accurate and ”0” otherwise. Two examples of annotated translation units are given bellow.

- **A correctly automatic annotated translation unit**
  - In a nutshell: the usage of the machinery for sifting, to loosen and rasp, or to prepare powdery substances and hygroscopic materials.
  - Kurz: Überall zum maschinellen Passieren, Auflockern und Raspeln oder zum Aufbereiten pulverförmiger Massen und hygroskopischer Materialien.
- **An incorrectly automatic annotated translation unit**
  - Large swimming pool and gym, for those who want to combine open air and relaxing activities with indoor training
  - Die Räume liegen direkt neben dem großen Pool und dem Fitnessraum, für all diejenigen die zu den vielzähligen Outdoor-Aktivitäten ein Trainigsprogramm in den Innenräumen kombinieren möchten.

In both examples the Hunalign score is higher than the fixed threshold but only the first example is correctly annotated automatically. The automatic annotator is a binary classifier and we can evaluate this classifier as is customary by comparing its annotation with a gold standard (the manual annotation). As one can see from the confusion matrix in table 1 the training set is imbalanced with only 27 percent negative examples.

| Measure | Value |
|---|---|
| Precision Positive class | 0.93 |
| Precision Negative class | 0.63 |
| Recall Positive class | 0.81 |
| Recall Negative class | 0.86 |
| F1 score Positive class | 0.87 |
| F1 score Negative class | 0.73 |
| Balanced Accuracy | 0.83 |

Table 2: Classification results

The precision, recall, F1-score and the balanced accuracy for the positive and negative classes are shown in table 2. All scores are high, showing that the heuristic based on Hunalign threshold is a good one. However, one should also consider that the automatically annotated set is not a representative sample of the test set provided by the organizers of the task. To have a representative sample much more translation units should have been annotated.

The annotation errors are mitigated by the fact that the Logistic regression classifier trained on the automatically annotated set will return the probability of the positive class. If the probability correlates with translation unit quality, then some translation units, even if not perfect, could be useful for training machine translation systems.

We counted some cases when the sentence in one language translates the sentence in the other language, but, at the same time, is more informative, as it contains another part for which there is no translation in the other language. Another worth making remark is the existence of many Bible passages, at least in the set we have manually annotated. They have lexical, morphological and syntactic characteristics which are specific to this kind of writing and which, when applied to other kinds of writing, will give inappropriate results. Although accepted as useful for MT in this task, they are probably good only for translating similar kinds of texts (i.e., religious ones).

A much better evaluation is provided by the task organizers. They have determined the quality of the cleaning performed by the teams by the BLEU score of a statistical machine translation (based on Moses) and a neural machine translation system (Marian) trained on two subsets as explained in the introduction section. There were 48 submissions and our system ranked in the range 22 - 31 depending on the subset and machine translation system used in the evaluation. For details regarding shared task preparation, the official results table and a survey of the methods used by the participating systems one should consult (Koehn et al., 2018).

## 4 Conclusions

In this paper we have presented a hybrid pipeline comprising rules and machine learning that was used to filter a noisy web English-German parallel corpus. The core of the pipeline is a logistic regression algorithm trained on an automatic annotated set. We have seen that the heuristic used to automatically annotate the training set is very good having 0.83 balanced accuracy (computed against the same set manually annotated). The pipeline also contains rules for re-scoring the translation units and a module based on cosine similarity to enhance the diversity of translation unit selection.

The core system, the manually annotated set and the python script for the evaluation procedure described in section 3 are publicly available on github[4].

## Acknowledgments

This study was supported by the Estonian Ministry of Education and Research (IUT20-56).

## References

Eduard Barbu, Carla Parra Escartín, Luisa Bentivogli, Matteo Negri, Marco Turchi, Constantin Orasan, and Marcello Federico. 2016. The first automatic translation memory cleaning shared task. *Machine Translation*, 30(3):145–166.

William A. Gale and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *COMPUTATIONAL LINGUISTICS*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830.

Jörg Tiedemann. 2011. *Bitext Alignment*. Number 14 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool, San Rafael, CA, USA.

D. Varga, L. Németh, P. Halácsy, A. Kornai, V. Trón, and V. Nagy. 2005. Parallel corpora for medium density languages. In *Recent Advances in Natural Language Processing (RANLP 2005)*, pages 590–596.

Friedel Wolff. 2016. Combining off-the-shelf components to clean a translation memory. *Machine Translation*, 30(3):167–181.

[4] https://github.com/SoimulPatriei/LogisticRegression-Shared-Task-Parallel-Corpus-Filtering.git

# Coverage and Cynicism: The AFRL Submission to the WMT 2018 Parallel Corpus Filtering Task

**Grant Erdmann, Jeremy Gwinnup**
Air Force Research Laboratory
grant.erdmann@us.af.mil,jeremy.gwinnup.1@us.af.mil

## Abstract

The WMT 2018 Parallel Corpus Filtering Task aims to test various methods of filtering a noisy parallel corpus, to make it useful for training machine translation systems. We describe the AFRL submissions, including their preprocessing methods and quality metrics. Numerical results indicate relative benefits of different options and show where our methods are competitive.

## 1 Introduction

For this task the participants were provided with a large corpus of parallel data in English and German. The corpus contains approximately $10^8$ lines, with approximately $10^9$ words in each language. Hunalign scores (Varga et al., 2005) also were provided for each line. The task organizers built statistical machine translation (SMT) and neural machine translation (NMT) systems from the scores produced, based on parallel training sets of $10^6$ and $10^7$ words.

Subset selection techniques often strive to reduce a set to the most useful. In this circumstance, this entails:

- Avoiding selecting a line with undue repetition of content of other selected lines. This can extend training times and/or skew the translation system to favor this type of line.
- Avoid selecting long lines, which will be ignored in training an NMT system.

In addition to adapting the corpus to the building of a general-purpose machine translation system, we must also deal with its significant noise. The main types of noise present in the given data are:

- Not natural language
- One or both languages are incorrect
- Correct languages and natural language, but not translations of each other

## 2 Preprocessing

As a first step, a rough preprocessing filter is applied to the data. This entails removing:

- Lines where either language contains more than 80 words
- Lines where either language contains less than 4 words
- Lines containing “www”, as lines with web addresses tend to provide less useful information
- Lines where the ratio of the number of English words to the number of German words is greater than three or less than one third
- Lines containing characters with the Unicode general category of “other”
- Lines where the English text is identical to the German text, after removing space, period, and numeric characters.
- Lines where numeric characters are different (or in a different order) in the two languages
- Lines where the hunalign score is less than 0.5 or greater than 1.5

The first of these criteria is based on limitations of NMT training, where long lines are discarded or truncated. The other criteria are highly empirical, based on indicators of apparent qualitative problems.

The remaining lines are put through further processing prior to scoring:

- Punctuation is normalized

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 872–876
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64102

- Words are truncated to 72 characters. The tokenizer attempts to separate German compound words, and long words cause it to hang.
- Language-specific tokenization is performed, using SYSTRAN's Linguistic Development Kit. Subword units are generated via byte-pair-encoding (BPE) (Gage, 1994). The BPE models are learned on a per-language basis, trained with 2000 byte-pair encoding merges, over all WMT 2018 news translation task parallel German–English data[1] without the Paracrawl[2] corpus. This small vocabulary was chosen to reduce the number of out-of-vocabulary tokens resulting from morphology and compounding.
- The BPE form is transformed into the format used for character-based processing, with denoted spaces and no subword continuation markers (e.g., `stand@@ ard prac@@ tice` becomes `stand ard _ prac tice`)
- Case features are removed, essentially allowing BPE formation using case but scoring lowercased.

This preprocessed text is used to generate the scores that determine a line's usefulness.

## 3 Coverage Metric

We use two metrics to estimate the relative appropriateness of a selected set to a reference. The first is our own coverage metric (Gwinnup et al., 2016), which we reproduce here. Let us select a subset $S$ from a larger set $C$ to maximize its similarity to a representative set $T$. Let our preferred subselected set size be $\tau$ times the size of $T$. Let $\mathcal{V}$ be a set of vocabulary elements of interest. Define $c_v(X)$ to be the count of the occurrence of feature $v \in \mathcal{V}$ in a given corpus $X$ and $c_v^\tau(T) = c_v(T)/\tau$ to be the scaled count that accounts for the preferred size of the selected set. The coverage $g$ is then given by

$$g(S,T,\tau) = \frac{\sum_{v\in\mathcal{V}} f(\min(c_v(S), c_v^\tau(T)))}{\sum_{v\in\mathcal{V}} f(c_v^\tau(T)) + p_v(S,T,\tau)} \quad (1)$$

where the oversaturation penalty $p_v(S,T,\tau)$ is

$$\max(0, c_v(S) - c_v^\tau(T)) \left[ f(c_v^\tau(T)+1) - f(c_v^\tau(T)) \right].$$

Here $f$ can be any submodular function, and we choose exclusively $f(x) = \log(1+x)$.

The final score reported for a line is the change it makes to the coverage metric on its inclusion. Lines which are not selected are given scores of zero.

## 4 Cynical Metric

As another approach we defined a metric based on the cynical selection method (Axelrod, 2017), which seeks to minimize the cross-entropy $H$. In our terms, this is

$$H(S,T) = -\sum_{v\in\mathcal{V}} \frac{c_v(T)}{\sum_{v'\in\mathcal{V}} c_{v'}(T)} \log \frac{c_v(S)}{\sum_{v'\in\mathcal{V}} c_{v'}(S)}. \quad (2)$$

We prefer to maximize metrics, so we define $h(S,T) = -H(S,T)$ as the cynical metric to maximize. Including the scaling factor $\tau$ would have no effect on the cross-entropy value.

Note that Axelrod (2017) defines the cross-entropy purely in terms of unigrams, motivated by an unsmoothed unigram language model. We include unigrams through 4-grams in our feature set $\mathcal{V}$. This extension to $n$-grams was not recommended by Axelrod (2017). However, we found it useful for this task.

The final score reported for a line is the change it makes to the cynical metric on its inclusion, with a maximum score of 1. Lines which are not selected are given scores of zero.

## 5 Set-building Algorithm

Whether the metric is our coverage metric or our cynical metric, the method of building the set is the same. We iterate the following two steps until the selected set is large enough:

1. Add the line that has the best effect on the metric.
2. Check if removing a line from the selected corpus would improve the metric. If so, remove the line with greatest such improvement, unless it was the most-recently selected or would lead to infinite cycling.

This is a greedy algorithm with review after each selection.

[1] http://www.statmt.org/wmt18/translation-task.html#download

[2] https://paracrawl.eu

## 6 Translation Score

The preceding processes and metrics were designed to remove many sources of error mentioned in the introduction of this paper. However, we have not yet dealt with the case of having both English and German lines being natural and useful, but the lines not being translations of one another. To help mitigate this phenomenon, we created a German–English NMT system using OpenNMT(Klein et al., 2017). It was trained on all WMT 2018 news translation task parallel German–English data, excluding the Paracrawl corpus. This system was a 4-layer bidirectional RNN, with 600-dimensional word embeddings and an RNN dimension of 1024, incorporating case features and a vocabulary from 2000 byte-pair encoding merges. The small vocabulary was chosen to reduce the number of out-of-vocabulary tokens resulting from morphology and compounding.

We translated all German the lines that survived the preprocessing step using this MT system. We computed the sentence-level Meteor scores (Denkowski and Lavie, 2011) of the English from the MT system, with the given data as the reference. We simply multiplied positive coverage or cynical scores by their Meteor scores.

## 7 Application

This section outlines the particulars of the method applied to the given data for this task. First, the Paracrawl data are preprocessed according to the method in §2. This reduces the set of potential lines from $10^8$ to $10^7$. This reduced set is divided into 100 parts of $10^5$ lines for scoring via batch processing.

Five different scoring methods will be considered. The baseline is `cvg-mix`, which uses our coverage metric and sums the coverage score for a small set ($\tau$ corresponding to $10^6$ total lines) and a large set ($\tau$ corresponding to $10^7$ total lines). Other scores are variants of this. The treatment `cvg-large` considers only the large set, and `cvg-small` considers only the small set. Meteor scores of translated lines are considered in `cvg-mix-meteor`. Finally, cynical scores are considered in `cyn-mix`.

## 8 Numerical Results

The results of the WMT 2018 Parallel Filtering Task are given by Bojar et al. (2018). BLEU scores for MT systems built from sets selected via our scoring methods are given in Tables 1-4. We do not consider the development set (newstest2017) in any analysis below, but we include it in the tables for completeness.

Several trends are apparent within our five submissions. First, including the Meteor score is always beneficial for the MT systems trained on smaller sets and rarely detrimental for the systems trained on larger sets. The filtering that includes a translation score, `cvg-mix-meteor`, is our top submission by mean BLEU score for all four MT systems. Second, the filter `cvg-small`, designed for producing a small training set, is poor at producing a large training set. Third, for the small training set there is almost always (test set EMEA in SMT excepted) a benefit from averaging the small training set method and the large training set method. Fourth, the coverage and cynical measures produce very similar results for SMT, but the cynical score is much better for the NMT system that used a small training set. The fact that selection methods differ in performance for SMT and NMT is known (van der Wees et al., 2017), but it is interesting that it is true for our two scoring methods.

Our best filtering method, `cvg-mix-meteor`, scores better than the mean performance of all non-AFRL methods in the task, for every test set and every MT system type. This method exhibits relatively better quality on the smaller ($10^6$-word) training sets, where it also bests the median. It is especially competitive with the top two systems using the $10^6$-word training sets on the test sets Acquis and KDE.

## 9 Conclusions

We have described a total of five different methods for filtering parallel data, as submitted to the WMT 2018 Parallel Filtering Task. We present numerical results, showing that our methods are especially competitive on certain test sets in the small training set condition.

Our coverage and cynical metrics yield approximately equivalent results in SMT, but the cynical metric is much better for the NMT system built on a small training set. Cynical scoring requires roughly half the computational time burden, so it is sometimes a good choice for NMT.

The ability to specify the size of the selected set is beneficial for our coverage scoring method in

Table 1: BLEU scores of created systems, $10^6$-word SMT. Filter mean excludes the development set (newstest2017). The two additional systems listed are the best performing in the task, by mean test set BLEU score. Set score statistics are over the 43 task submissions from other participants.

| Filter name | newstest2017 | newstest2018 | iwslt2017 | Acquis | EMEA | Global Vcs | KDE | mean |
|---|---|---|---|---|---|---|---|---|
| cvg-mix | 20.61 | 25.22 | 18.39 | 17.65 | 23.64 | 19.35 | 21.12 | 20.89 |
| cvg-small | 20.38 | 25.03 | 18.04 | 15.82 | 24.31 | 18.99 | 20.46 | 20.44 |
| cvg-large | 20.39 | 25.00 | 17.97 | 15.81 | 24.30 | 18.98 | 20.43 | 20.42 |
| cyn-mix | 20.52 | 25.45 | 18.44 | 17.22 | 23.72 | 19.16 | 21.10 | 20.85 |
| cvg-mix-meteor | **21.46** | **26.41** | **19.01** | **17.98** | **24.55** | **19.90** | **22.06** | **21.65** |
| microsoft | 24.04 | 28.18 | 20.39 | 17.13 | 26.95 | 21.20 | 22.76 | 22.77 |
| rwth-nn-redundant | 24.36 | 28.40 | 20.60 | 18.58 | 26.12 | 21.37 | 21.48 | 22.76 |
| median | 21.77 | 24.91 | 18.50 | 16.11 | 23.99 | 18.98 | 21.48 | 20.66 |
| mean | 20.57 | 23.70 | 17.30 | 14.70 | 22.71 | 18.14 | 20.65 | 19.53 |
| std dev | 3.73 | 4.81 | 3.93 | 3.31 | 3.79 | 3.40 | 2.91 | 3.69 |

Table 2: BLEU scores of created systems, $10^7$-word SMT. Filter mean excludes the development set (newstest2017). The two additional systems listed are the best performing in the task, by mean test set BLEU score. Set score statistics are over the 43 task submissions from other participants.

| Filter name | newstest2017 | newstest2018 | iwslt2017 | Acquis | EMEA | Global Vcs | KDE | mean |
|---|---|---|---|---|---|---|---|---|
| cvg-mix | 23.36 | 28.61 | **21.24** | **18.67** | 28.05 | 21.49 | 23.68 | 23.62 |
| cvg-small | 21.10 | 25.88 | 18.98 | 18.19 | 24.06 | 20.06 | 20.97 | 21.36 |
| cvg-large | **23.40** | **28.76** | 21.11 | 18.61 | 28.04 | 21.55 | 23.75 | 23.64 |
| cyn-mix | 23.19 | 28.28 | 21.06 | 18.49 | 27.94 | 21.26 | 23.59 | 23.44 |
| cvg-mix-meteor | 23.33 | 28.68 | 21.12 | 18.66 | **28.22** | **21.66** | **23.85** | **23.70** |
| microsoft | 24.48 | 29.99 | 21.98 | 19.43 | 29.81 | 22.63 | 24.67 | 24.75 |
| prompsit-al | 24.50 | 29.83 | 21.67 | 19.71 | 29.48 | 22.54 | 24.72 | 24.66 |
| median | 23.96 | 29.26 | 21.52 | 19.19 | 28.89 | 22.15 | 24.33 | 24.22 |
| mean | 22.85 | 27.91 | 20.46 | 18.18 | 27.67 | 21.27 | 23.65 | 23.19 |
| std dev | 2.91 | 3.61 | 2.70 | 2.55 | 3.21 | 2.46 | 1.99 | 2.75 |

Table 3: BLEU scores of created systems, $10^6$-word NMT. Filter mean excludes the development set (newstest2017). The two additional systems listed are the best performing in the task, by mean test set BLEU score. Set score statistics are over the 43 task submissions from other participants.

| Filter name | newstest2017 | newstest2018 | iwslt2017 | Acquis | EMEA | Global Vcs | KDE | mean |
|---|---|---|---|---|---|---|---|---|
| cvg-mix | 15.16 | 18.81 | 10.36 | 20.97 | 25.04 | 14.06 | 20.84 | 18.35 |
| cvg-small | 8.11 | 10.40 | 5.28 | 13.20 | 22.18 | 8.08 | 15.40 | 12.42 |
| cvg-large | 8.42 | 10.70 | 5.80 | 13.31 | 22.27 | 8.43 | 15.92 | 12.74 |
| cyn-mix | 22.35 | 28.06 | 20.35 | 21.44 | 27.29 | 21.49 | 22.03 | 23.44 |
| cvg-mix-meteor | **26.43** | **32.03** | **22.01** | **22.50** | **28.01** | **24.10** | **22.89** | **25.26** |
| microsoft | 27.22 | 34.32 | 23.86 | 20.87 | 30.75 | 25.46 | 25.47 | 26.79 |
| rwth-nn-redundant | 28.08 | 34.65 | 23.96 | 22.01 | 29.23 | 25.38 | 21.50 | 26.12 |
| median | 24.04 | 29.90 | 20.53 | 18.46 | 25.71 | 22.42 | 21.50 | 23.09 |
| mean | 21.21 | 26.25 | 18.20 | 16.07 | 23.42 | 19.74 | 19.07 | 20.46 |
| std dev | 6.93 | 8.80 | 6.64 | 5.75 | 6.82 | 6.46 | 6.41 | 6.81 |

Table 4: BLEU scores of created systems, $10^7$-word NMT. Filter mean excludes the development set (newstest2017). The two additional systems listed are the best performing in the task, by mean test set BLEU score. Set score statistics are over the 40 task submissions from other participants.

| Filter name | newstest2017 | newstest2018 | iwslt2017 | Acquis | EMEA | Global Vcs | KDE | mean |
|---|---|---|---|---|---|---|---|---|
| cvg-mix | 28.76 | 36.00 | 24.81 | 22.94 | 32.88 | **26.89** | 26.13 | 28.27 |
| cvg-small | 17.00 | 22.33 | 15.95 | 19.71 | 24.31 | 18.17 | 16.77 | 19.54 |
| cvg-large | 28.81 | 35.63 | **25.10** | **23.20** | 33.06 | 26.75 | 26.13 | 28.31 |
| cyn-mix | 28.04 | 34.82 | 23.85 | 22.78 | 32.91 | 26.21 | 25.68 | 27.71 |
| cvg-mix-meteor | **28.98** | **36.07** | 24.79 | 23.19 | **33.15** | 26.84 | **26.29** | **28.39** |
| microsoft | 31.04 | 38.39 | 26.06 | 24.91 | 34.68 | 28.04 | 28.37 | 30.07 |
| alibaba-div | 30.55 | 38.02 | 25.71 | 25.03 | 34.65 | 27.90 | 28.33 | 29.94 |
| median | 29.47 | 36.84 | 25.19 | 24.17 | 33.46 | 27.00 | 27.44 | 29.02 |
| mean | 26.25 | 32.72 | 22.17 | 21.42 | 30.63 | 24.40 | 25.29 | 26.11 |
| std dev | 7.47 | 9.50 | 6.67 | 6.15 | 6.77 | 6.36 | 5.37 | 6.80 |

the small training set conditions, where it yields about the same results for an order of magnitude less computation time. Unfortunately, specifying a desired output set size is not as obvious for cynical scoring.

Inclusion of a translation metric score such as Meteor is beneficial, and the simplistic version given here produced our best system. Introducing of a translation metric score directly in the set-building process would help in avoiding redundancy.

Optimizing the heuristic and empirical prefiltering and preprocessing steps given here could yield substantial benefit. We have doubtlessly removed some beneficial lines in the prefiltering, which excluded up to 90% of the data. In fact, the prefiltering could conceivably be replaced by moving the application of the machine translation system to before scoring, rather than after. Unfortunately this change would cause much more of a computational burden, as every line would need to be translated.

## References

Amittai Axelrod. 2017. Cynical selection of language model training data. *Computing Research Repository*, arXiv:1709.02279. Version 1.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12:23–38.

Jeremy Gwinnup, Tim Anderson, Grant Erdmann, Katherine Young, Michaeel Kazi, Elizabeth Salesky, and Brian Thompson. 2016. The AFRL-MITLL WMT16 news-translation task systems. In *Proceedings of the First Conference on Machine Translation*, pages 296–302, Berlin, Germany. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Dániel Varga, László Németh, Péter Halácsy, András Kornai, Viktor Trón, and Viktor Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of the Recent Advances in Natural Language Processing RANLP 2005*, pages 590–596, Borovets, Bulgaria.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

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 8 Aug 2018. Originator reference number RH-18-118707. Case number 88ABW-2018-3956.

# Webinterpret Submission to the WMT2018 Shared Task on Parallel Corpus Filtering

**Marina Fomicheva**[*]
AT Language Solutions
mari.fomicheva@gmail.com

**Jesús González-Rubio**
WebInterpret
jesus.g.rubio@gmail.com

## Abstract

This paper describes the participation of Webinterpret in the shared task on parallel corpus filtering at the Third Conference on Machine Translation (WMT 2018). The paper describes the main characteristics of our approach and discusses the results obtained on the data sets published for the shared task.

## 1 Task description

Parallel corpus filtering task at WMT18[1] tackles the problem of cleaning noisy parallel corpora. Given a noisy parallel corpus (crawled from the web), participants develop methods to filter it to a smaller size of high quality sentence pairs.

Specifically, the organizers provide a very noisy 1 billion word German–English corpus crawled from the web as part of the Paracrawl project[2]. Participants are asked to select a subset of sentence pairs that amount to (a) 100 million words, and (b) 10 million words. The quality of the resulting subsets is determined by the quality of a statistical and a neural Machine Translation (MT) systems trained on the selected data. The quality of the translation systems is measured computing the BLEU score on the (a) official WMT 2018 news translation test set and (b) another undisclosed test set.

The organizers make explicit that the task addresses the challenge of *data quality* and *not domain-relatedness* of the data for a particular use case. Hence, they discourage participants from sub-sampling the corpus for relevance to the news domain despite being one of the evaluation test sets. Organizers thus place more emphasis on the second undisclosed test set, although they report both scores.

The provided raw parallel corpus is the outcome of a processing pipeline that aimed for high recall at the cost of precision, which makes it extremely noisy. The corpus exhibits noise of all kinds (wrong language in source and target, sentence pairs that are not translations of each other, bad language, incomplete or bad translations, etc.).

We address this problem under the framework of quality estimation (QE) (Blatz et al., 2004). QE aims at assessing MT quality in the absence of reference translation, based on the features extracted from the source sentence and from the MT output. We consider parallel corpus filtering as a QE task where the goal is to estimate to what extent a pair of sentences in two languages correspond and, therefore, can be considered as translations of each other.

The rest of this paper is organized as follows. First, we describe our submission. Next, we present our experiments and the results of the shared task. Finally, we close the paper with the conclusions and some ideas for future work.

## 2 Corpus filtering as QE task

We frame the corpus filtering task within the QE framework. Given a pair of sentences $(\mathbf{s}, \mathbf{t})$, we first compute a set of features indicating to what extent the sentences correspond to each other. Then, these features are used to predict a binary score indicating if the sentences in the pair can be considered translations of each other.

In order to make the training process effective, any binary classification model needs to use both positive and negative examples. In our context positive examples are pairs of original and translated sentences, whereas negative examples are sentence pairs that cannot be considered transla-

[*] Marina Fomicheva worked at Webinterpret at the time of preparation of this submission.

[1] http://www.statmt.org/wmt18/parallel-corpus-filtering.html

[2] https://paracrawl.eu/

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 877–881
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64103

tions of each other. Positive examples can be easily obtained from clean parallel corpora, and, while there is no explicit corpus with negative examples, these can be generated on demand.

We use the confidence score from our binary classifier as the final score for our submission to the shared task. As described in the previous Section, based on this score, the sentence pairs in the original noisy corpus provided by the organizers will be sorted and then the first N pairs will be selected and used to train the MT systems.

Note that this approach may be sub-optimal since it considers each individual pair of sentences in isolation from the rest. In exchange for this, we end up with a much more efficient method, linear in the size of the noisy data.

Next, we describe in detail the features we used for our submission (Sec. 2.1), the process we followed for generating negative examples (Sec. 2.2) and the classification model we chose (Sec. 2.3).

### 2.1 Features

We use a rich variety of features intended to capture what it means to be an adequate training pair of sentences. For simplicity, we split them into three categories.

**Adequacy** These features measure how much of the meaning of the original is expressed in the translation and vice versa. We use probabilistic lexicons with different formulations of word alignment to estimate the extent to which the words in the original and translated sentences correspond to each other.

- *Average Max lexical probability (2 f.)*: originally proposed by (Ueffing and Ney, 2005) for word-level QE. It measures the average maximum probability of translation for each word in the sentence. We apply it in both source-to-target and target-to-source directions. Formally, source-to-target is given by:

$$\frac{1}{n}\sum_{1}^{n}\max_{j=0}^{m} P(t_i \mid s_j)$$

where the source word $\mathbf{s} = s_1 \ldots s_m$ has $m$ words, the target sentence $\mathbf{t} = t_1 \ldots t_n$ has $n$ words and the word $s_0$ indicates the NULL word (Brown et al., 1993). For target-to-source, source and target words swap their roles.

- *Cross-entropy (2 f.)*: proposed by (Xu and Koehn, 2017), it measures a "distance" between the sentence pairs based on a bag-of-words translation model. Specifically, the "distance" is measured as the cross-entropy between the bag-of-words of the actual sentence and the bag-of-words estimated from the other sentence in the pair via the probabilistic lexicon. We apply it in both source-to-target and target-to-source directions.

**Fluency** This type of features aim at capturing if the sentences are well-formed grammatically, contain correct spellings, adhere to common use of terms, titles and names, are intuitively acceptable and can be sensibly interpreted by a native speaker. We use two different features, both based on language models:

- *Language model score (2 f.)*: given language models for the source and target languages, we use as features the log probability of each sentence in the pair computed with the corresponding model.

- *Perplexity (2 f.)*: is measured as the inverse probability of the sentence normalized by its number of words. Again, we apply it to both source and target sentences in the pair.

**Shape features** These features can be seen as an extension of adequacy since they measure the mismatch between the frequency of different tokens between the two sentences in the pair; these features are quite commonly used in the QE literature, (Specia et al., 2015) *inter alia*.

- *Counts (8 f.)*: count of words, numbers, alphanumeric tokens, and punctuation in both source and target sentences.

- *Jaccard index (4 f.)*: metric that measures the similarity and diversity of the sets of tokens between the source and target sentences. Formally it is defined as:

$$\frac{\mid A \cap B \mid}{\mid A \cup B \mid}$$

where $A$ and $B$ are the set of tokens of the source and target sentences respectively. We apply it to words, numbers, alphanumeric tokens and punctuation.

- *Counts difference (16 f.)*: we compute four metrics from the counts of tokens: the ratio in both directions, the absolute difference, and the absolute difference normalized by the maximum number of tokens of both sentences. Each of these metrics is applied to four different types of tokens: words, numbers, alphanumeric tokens and punctuation.

- *Specific punctuation (12 f.)* same as the previous features, but in this case we only compute the absolute difference and the normalized difference for specific punctuation tokens: dot (.), comma (,), colon (:), semicolon (;), exclamation mark (!), and question mark (?).

### 2.2 Training regime

An important consideration for this task is how to obtain suitable examples to train the classification model. Positive examples are easy to obtain since any clean parallel corpus provide us with plenty of them. Negative examples, however, are not readily available -there exist no collection of “wrong” sentence pairs. Fortunately, they can be easily generated on demand. We mostly followed the approach described in (Xu and Koehn, 2017), perturbing one or both of the sentences in a pair to create a new synthetic pair that by construction constitutes a negative example.

We apply three different perturbation operations when generating negative pairs:

- *Swap*: exchange source and target sentences.

- *Copy*: two copies of the same string. We apply it to both source and target strings.

- *Randomization*: replace the source or target sentence by a random sentence from the same side of the corpus.

As can be seen from above, we focus on the perturbation operations that mess with the correct alignment between the sentences. Thus, we aim at identifying correctly aligned sentence pairs. A complementary approach would be to aim at detecting the actual “quality” of the sentence pair, or, in other words, how valuable a sentence pair is when used for training MT systems. However, this is left for future developments.

### 2.3 Classification model

We did some initial experiments testing the performance of different classifiers on the task of distinguishing between actual original-translation sentence pairs and the synthetically generated negative examples (see Sec. 3.2 for details on the data we used). Gradient boosting algorithm (Friedman, 2002) obtained the highest accuracy and, therefore, we used it for our final submission.

Gradient boosting (Gra) is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. Similar to other boosting methods, it builds the models in a stage-wise fashion and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

## 3 Submission

Next, we describe the tools and the data we exploited for feature extraction, the data used to train the classifier, and the results of our participation in the shared task.

### 3.1 Feature Extraction

We need to generate two types of models to extract our features: probabilistic lexicons and language models. We used the probabilistic lexicons that can be obtained as a sub product of the training of full statistical models. In particular, we used Moses (Koehn et al., 2007) with its default configuration with the News Commentary V13 parallel corpus as provided for the News translation shared task. We used the same corpora to train the language models. For this, we used Kenlm (Heafield et al., 2013) and estimated models of order 5.

### 3.2 Training the classifier

We also used News Commentary V13 parallel corpus for training the classifier. We generated as many negative examples as positive sentence pairs in the corpus for a total of almost 600k data points. The negative examples were evenly distributed among the three perturbation operations described in the previous section. We used the implementation of gradient boosting classifier from the scikit-learn library[3] to train our model. The model was then applied to each sentence pair in the noisy Paracrawl corpus from the shared task.

[3] http://scikit-learn.org/stable/index.html

We used the probability of the positive class as predicted by the classifier as the final scores in our submission.

We also conducted some initial experiments using the Common Crawl corpus, under the rationale that it would be closer to the domain of the noisy data from the Paracrawl corpus. However, Common Crawl data has quite a large number of misaligned sentences. To handle this issue we implemented an iterative training process which comprises the following steps: a) train the model using all available data as positive class and synthetically generated data as negative class (see Sec 2.2); b) use the trained model to clean the available data eliminating the sentence pairs assigned to the negative class with a very high probability; c) use the cleaned data to train a new model; d) repeat until no more sentence pairs can be eliminated with a given threshold. An advantage of this approach is that it allows to be less dependent on the quality of the initial training data. However, we had to stop exploring this direction due to time constraints.

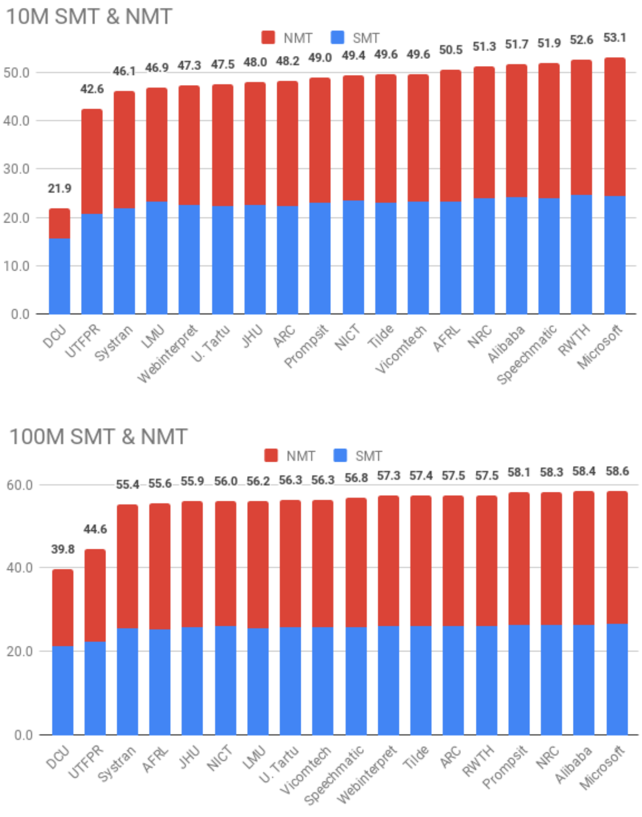


Figure 1: Best submission of each participant institution. We display BLEU [%] results stacked for SMT (blue) and NMT (red).

### 3.3 Evaluation and results

Participants in the shared task have to submit a file with quality scores, one per line, corresponding to the sentence pairs on the 1 billion word German-English Paracrawl corpus. Scores do not have to be meaningful, except that higher scores indicate better quality. The performance of the submissions is evaluated by sub-sampling 10 million and 100 million word corpora based on these scores, training statistical (Koehn et al., 2007) and neural (Junczys-Dowmunt et al., 2018) MT systems with these corpora, and assessing translation quality on six blind test sets[4] using the BLEU (Papineni et al., 2002) score.

Figure 1 displays the score of the best submission of each individual participant institution. The top plot shows the results for the 10 million token sub-sampled corpus, and the bottom plot shows the results for the 100 million token corpus. Scores are the aggregation of the BLEU scores of the statistical and neural systems averaged over the six blind test sets.

One first observation we can make is that (almost) all scores are quite close to each other with little variation between them; particularly in the 100 million condition. Also, the scores for the statistical and neural systems tend to follow the same pattern. We do not have confidence intervals available which makes difficult to interpret the observed differences between systems. Still, in the case of 100 million tokens sub-sampling, it seems quite clear that all the systems except for the DCU and UTFPR submissions are of the same quality. There is only a 5% relative improvement between the last system of this group and the best submission to the task. Scores are a bit more spread out in the 10 million tokens sub-sampling. This indicates that 100 million sample neutralizes the differences between the data cleaning methods and allows (almost) all systems to reach a theoretical maximum.

Our submission (Webinterpret) scored 22.5 for statistical and 24.8 for neural MT systems on the 10 million tokens sub-sampling, in comparison to the corresponding scores of 24.5 and 28.6 achieved by the best submission. For the 100 million condition, we scored 26.1 and 31.2, in comparison to the best system with the respective scores of 26.5 and 32.1.

[4] Tests: newstest 2018, iwslt 2017, Acquis, EMEA, Global Voices, and KDE.

## 4 Conclusions

We have presented our submission to the WMT18 shared task on parallel corpus filtering. We frame the task as a QE problem, where we estimate how well two sentences correspond to each other to be part of a training sample for MT models. Our approach is computationally light, takes advantage of well-known methods used for QE, and exploits a general training regime that allows to customize it by defining under demand samples of negative examples.[5]

There are several directions that can be explored to extend this approach:

- Use a neural model to automatically estimate the features relevant for the system instead of hand-crafting them.
- Extend the training regime with new perturbation operations, in particular those that degrade the quality of the pair so it is less valuable as training data for MT.
- Implement an iterative training procedure where steps of model training and data cleaning are repeated over the available training data until convergence. This will make training more robust and less dependent on the quality of available training data.

## Acknowledgments

Work funded by WebInterpret.

## References

Gradient boosting. https://en.wikipedia.org/wiki/Gradient_boosting. Accessed: 2018-08-15.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321. Association for Computational Linguistics.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311.

Jerome H. Friedman. 2002. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4):367–378.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

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. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL 2007, System Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *ACL-IJCNLP 2015 System Demonstrations*, pages 115–120.

Nicola Ueffing and Hermann Ney. 2005. Application of word-level confidence measures in interactive statistical machine translation. In *Proceedings of the European Association for Machine Translation conference*, pages 262–270.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950. Association for Computational Linguistics.

[5] The code developed to prepare our submission is available at https://github.com/mfomicheva/parallel_data_cleaning.

# An Unsupervised System for Parallel Corpus Filtering

**Viktor Hangya** and **Alexander Fraser**
Center for Information and Language Processing
LMU Munich, Germany
{hangyav, fraser}@cis.lmu.de

## Abstract

In this paper we describe LMU Munich's submission for the *WMT 2018 Parallel Corpus Filtering* shared task which addresses the problem of cleaning noisy parallel corpora. The task of mining and cleaning parallel sentences is important for improving the quality of machine translation systems, especially for low-resource languages. We tackle this problem in a fully unsupervised fashion relying on bilingual word embeddings created without any bilingual signal. After pre-filtering noisy data we rank sentence pairs by calculating bilingual sentence-level similarities and then remove redundant data by employing monolingual similarity as well. Our unsupervised system achieved good performance during the official evaluation of the shared task, scoring only a few BLEU points behind the best systems, while not requiring any parallel training data.

## 1 Introduction

Machine translation is important for eliminating language barriers in everyday life. To train systems which can produce good quality translations large parallel corpora are needed. Mining parallel sentences from various sources in order to train better performing MT systems is essential, especially for low resource languages. Previous efforts[1] showed that it is possible to crawl parallel data from the web, but also showed that additional steps are necessary to filter noisy sentence pairs. In this paper we introduce our approach to filter noisy parallel corpora without the need of any initial bilingual signal to train the filtering system.

We participate in the *WMT 2018 Parallel Corpus Filtering* shared task with our system which tackles the problem of selecting the best quality sentence pairs for training both statistical and neural MT systems (Koehn et al., 2018). A lot of previous work has studied the problem of parallel data cleaning. Esplà-Gomis and Forcada (2010) proposed BiTextor which filters data based on sentence alignment scores and URL information. Similarly, word alignments and language modeling were used in (Denkowski et al., 2012) to select sentence pairs that are useful for training an MT system. Xu and Koehn (2017) proposed Zipporah, a logistic regression based model that uses bag-of-words translation features to measure fluency and adequacy in order to score sentence pairs. Another line of work is to select data based on the target domain. A static sentence-selection method was used for domain adaptation based on the internal sentence embedding of NMT (Wang et al., 2017) while van der Wees et al. (2017) used domain-based cross-entropy as a criterion to gradually fine-tune the NMT training in a dynamic manner. In contrast with previous work, we do not rely on any bilingual supervision, making our approach applicable to language pairs which lack initial parallel resources. Similarly to the work of Kajiwara and Komachi (2016), where word embeddings were used to mine monolingual sentence pairs for text simplification, we use a word level metric to compute sentence pair similarity in a computationally efficient way.

Our approach consists of three steps. Due to the noisiness of the input data we use a pre-filtering step which detects sentences which are not useful. We developed a simple rule-based method which looks for sentence pairs which for example came from the wrong languages or have significantly different lengths. As a second step, we calculate sentence pair similarities using bilingual word embeddings and orthographic information. In the third step, we perform post-ranking where we counterweight source language sentences which

[1] https://paracrawl.eu

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 882–887
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64104

are less fluent or redundant using language modeling and monolingual document similarity respectively. Our system is fully unsupervised, i.e., we do not use any parallel data for the training of our methods. We show results on the official test sets of the shared task which includes six datasets from different sources. Although, our method is fully unsupervised it achieves good performance on the extrinsic task of training MT systems on the filtered parallel data, scoring only 2.17 BLEU points behind the best systems.

## 2 Approach

In this section we introduce our approach for the filtering task. Since parallel sentence mining is most crucial for resource-poor languages our goal was to develop a system that does not need any bilingual signal for training. Our approach is based on recent developments in the field of bilingual word embeddings, i.e., it was shown that good quality bilingual embeddings can be trained using only source and target language monolingual data (Conneau et al., 2017). As was mentioned in the previous section our approach consists of three steps which we introduce below. In each step we score the input candidate sentence pairs which are used at the sampling step to select sentence pairs before the training of MT systems. Higher score means higher probability for being selected during the sampling process. For more detail about the data, the preprocessing and the sampling procedure see section 3.

### 2.1 Pre-Filtering

The input data, released by the shared task organizers, contain a large amount of erroneous candidate sentence pairs which can be filtered out based on some simple heuristics. For detecting these instances we use the following rules and set the weight of these noisy candidate pairs to zero. Note, we ignore the candidates selected here in later steps for reasons of speed.

1. **Hunalign** scores of the sentence pairs were released with the data. We ignore candidates if the initial score is less then $0.0$.

2. If either of the sentences has a **length** of less then 3 tokens we consider it as noise.

3. A good indicator of bad alignment of sentences is their **length difference**. If this value is greater than $15$ tokens we set its weight to zero.

4. We also consider a candidate as noise if the **number and URL ratio**, compared to the number of all tokens, is greater than $0.6$.

5. In many cases the **language** of the sentences is incorrect. We use the system of Sarwar et al. (2001) to detect these instances.

### 2.2 Scoring

In the main step of our approach we calculate the score of a candidate sentence pair based on the similarities of the contained words. First, we describe how we train bilingual word embeddings and then we describe the method for sentence similarity.

**Bilingual word embeddings** Recently, Conneau et al. (2017) showed that good quality bilingual embeddings can be produced by training monolingual word embedding spaces for both source and target languages and mapping them to a shared space without any bilingual signal. We follow this approach and use bilingual word embeddings, trained in an unsupervised fashion. For this we use the system released by (Conneau et al., 2017). We discuss the used data and parameters in section 3.

**Sentence pair similarity** Given a candidate pair of source and target sentences $S$ and $T$, the similarity score is calculated by iterating over the words in $S$ from left to right and pairing each word $s \in S$, in a greedy fashion, with the word $t \in T$ that has the highest cosine similarity based on our dictionary. We then greedily eliminate $t$ from $T$, so that it cannot be matched by a later word “s”. Then, the averaged word-pair similarity gives the final score. We remove stopwords, digits and punctuation from texts before calculating similarity. Note, this idea is similar to *Word Movers Distance* introduced in (Kusner et al., 2015) but simpler due to runtime considerations on huge corpora.

As was shown in previous work (Braune et al., 2018), the quality of bilingual word similarity can be significantly improved by using orthographic cues, especially for rare words. We extend this idea to the sentence level by using a dictionary containing orthographically similar source-target language word pairs and their similarity. We define orthographic similarity as one minus the normalized Levenshtein distance. We use this orthographic dictionary together with the BWE-based dictionary when mining parallel sentences by using the higher value from the two dictionaries. If the given word pair is not in a dictionary we consider their similarity as $0.0$ for that dictionary. One issue with orthographic similarity of words is that it tends to give high scores to sentences which contain many orthographically similar words, e.g., a sentence with a list of named entities, which are often not useful for MT systems. To overcome this issue, we multiply the orthographic word similarities with $0.2$.

### 2.3 Post-Ranking

In the third step we re-rank candidates from the previous step in order to reduce the number of redundant sentence pairs and to ensure that we have more fluent sentences. We apply these steps only to the source sentences due to speed considerations.

**Monolingual Document Similarity** The input corpus contains redundant sentences, i.e., sentences which have similar structure and meaning, and which are often generated based on predefined sentence templates. It is enough to use only one element from these clusters of redundant sentences since the rest does not have a big impact on the translation quality. Due to the huge size of the input data we used a simple thus fast approach to detect redundant sentences and decrease their score. First, we embed each source side sentence to a fixed sized sentence embedding by simply averaging the word embeddings of the words in the sentence. We calculate sentence similarities of each possible pairs which can be done efficiently even for large inputs (Johnson et al., 2017). We use cosine as the similarity metric and we consider those sentences as redundant which have lower difference than $0.02$ between the similarity value of its top two most similar sentences. We multiply the original score of redundant sentences by $0.5$.

**Language model** It is beneficial to use fluent sentences for training MT systems. To take this aspect into consideration we used KenLM language model (Heafield et al., 2013) to change the score of a candidate pair based on the source side sentence's normalized language model probability. We multiply scores if the given sentence has higher (lower) probability than $1 \times 10^{-3}$ ($5 \times 10^{-6}$) by $1.5$ ($0.5$).

## 3 Experimental Setup

The goal of the shared task is, given a noisy parallel corpus, to filter candidate sentence pairs that are most useful for training MT systems. Candidate pairs have to be scored based on the predicted quality of the corresponding candidate where the scores do not have a special meaning except that higher values indicate better quality. To produce the actual training data for the MT systems the scored corpus is sampled using an official tool, released by the organizers, which samples sentences with a probability proportional to their scores.

### 3.1 Data

A German-English dataset was released containing 1 billion (English) tokens. The corpus was crawled from the web as part of the *ParaCrawl* project. After extracting texts from web pages with BiTextor (Esplà-Gomis and Forcada, 2010), documents and sentences were aligned using (Buck and Koehn, 2016) and Hunalign (Varga et al., 2007) respectively. The aligned sentence pairs are the candidates which have to be scored for the sampling process and used as training parallel data for the MT systems. The alignment scores of the candidate sentence pairs were also released which do not by themselves correlate strongly with sentence pair quality which we show in section 4. For more details of the data see the overview paper of the shared task (Koehn et al., 2018). As an additional data source we use monolingual German and English NewsCrawl sentences from the time period between 2011 and 2014 (Bojar et al., 2014) which we use to train word embeddings and the language model.

### 3.2 Evaluation

To evaluate systems two setups were performed: (i) sampling 10M tokens and (ii) 100M tokens from the scored corpus using the released sampler tool. The quality of the resulting subsets is determined by the quality of a German-English SMT (Koehn et al., 2007) and an NMT (Junczys-Dowmunt et al., 2018) system trained on this data and using BLEU to measure translation quality. We will refer to these setups as SMT 10M, SMT 100M, NMT 10M and NMT 100M. As development set newstest 2017 was used, while newstest

| | | newstest 2017 | newstest 2018 | iwslt2017 | Acquis | EMEA | Global Voices | KDE | avg |
|---|---|---|---|---|---|---|---|---|---|
| SMT 10M | lmu-ds-lm | 21.73 | 28.03 | 20.61 | 17.97 | 26.95 | 21.45 | 24.73 | 23.29 |
| | lmu-ds | 21.71 | 28.03 | 20.57 | 17.96 | 26.96 | 21.46 | 24.58 | 23.26 |
| | lmu | 19.62 | 25.35 | 19.67 | 15.30 | 25.32 | 20.03 | 23.08 | 21.46 |
| SMT 100M | lmu-ds-lm | 24.86 | 30.14 | 22.42 | 21.47 | 30.08 | 23.09 | 26.20 | 25.57 |
| | lmu-ds | 24.86 | 30.00 | 22.31 | 21.25 | 30.18 | 23.19 | 26.11 | 25.51 |
| | lmu | 25.09 | 30.34 | 22.37 | 20.98 | 30.44 | 23.27 | 26.24 | 25.61 |
| NMT 10M | lmu-ds-lm | 26.17 | 31.89 | 22.40 | 18.51 | 27.01 | 24.60 | 17.46 | 23.65 |
| | lmu-ds | 26.22 | 31.79 | 22.09 | 18.43 | 27.14 | 24.53 | 17.94 | 23.65 |
| | lmu | 23.03 | 28.79 | 21.06 | 16.01 | 26.98 | 23.30 | 21.64 | 22.96 |
| NMT 100M | lmu-ds-lm | 29.14 | 36.99 | 25.48 | 25.19 | 33.46 | 27.52 | 28.17 | 29.47 |
| | lmu-ds | 29.33 | 36.71 | 25.48 | 25.25 | 34.15 | 27.67 | 27.95 | 29.54 |
| | lmu | 30.82 | 37.78 | 25.95 | 25.77 | 35.61 | 28.48 | 29.62 | 30.54 |

Table 1: BLEU scores of our setups on the different datasets. We underline best results for each setup and dataset.

| Insitution | SMT 10M | SMT 100M | NMT 10M | NMT 100M |
|---|---|---|---|---|
| RWTH | 24.58 | 26.21 | 28.01 | 31.29 |
| Microsoft | 24.45 | 26.50 | 28.62 | 32.06 |
| Alibaba | 24.11 | 26.44 | 27.60 | 31.93 |
| NRC | 23.89 | 26.40 | 27.41 | 31.88 |
| Speechmatics | 23.88 | 25.85 | 27.97 | 31.00 |
| NICT | 23.46 | 25.98 | 25.94 | 30.04 |
| AFRL | 23.36 | 25.32 | 27.09 | 30.28 |
| Vicomtech | 23.29 | 25.91 | 26.35 | 30.40 |
| LMU | 23.29 | 25.61 | 23.65 | 30.54 |
| Tilde | 23.03 | 26.19 | 26.56 | 31.24 |
| Prompsit | 22.94 | 26.41 | 26.05 | 31.83 |
| ARC | 22.68 | 26.13 | 25.79 | 31.34 |
| JHU | 22.61 | 25.84 | 25.41 | 30.16 |
| MAJE | 22.53 | 26.07 | 24.81 | 31.20 |
| Univ. Tartu | 22.31 | 25.70 | 25.17 | 30.60 |
| Systran | 21.83 | 25.44 | 24.30 | 29.91 |
| UTFPR | 20.81 | 22.35 | 21.75 | 22.23 |
| DCU | 15.67 | 21.19 | 6.27 | 18.60 |

Table 2: Best systems of participants on the four setups averaged over all test sets.

2018, iwslt2017, Acquis, EMEA, Global Voices and KDE were the undisclosed test sets (Koehn et al., 2018).

### 3.3 Parameter setup

We preprocessed all data using the tokenizer from Moses with aggressive mode (Koehn et al., 2007) and lower casing. To train monolingual word embeddings we used FastText (Bojanowski et al., 2016) with default parameters except the dimension of the vectors which is 300. As input the concatenation of the shared task data and NewsCrawl was used. For the unsupervised mapping we ran (Conneau et al., 2017) using the source and target language monolingual spaces. As a language model we used KenLM (Heafield et al., 2013), with n-gram size 5 and using default values for the rest of the parameters, on the source side of our data. All other parameters introduced earlier are based on manual analysis of the data and non-exhaustive tuning on the development set. During development we only run SMT 10M due to time constraints.

## 4 Results

We present official BLEU scores of our systems on the four setups and seven datasets in table 1. Our default system *lmu* applies pre-filtering and scoring and we incrementally add monolingual document similarity and language modeling post-ranking steps. During development we calculated the performance of only applying the pre-filtering step on newstest 2017 with SMT 10M which resulted in a score of 15.53 BLEU while the released hunalign scores resulted in a score of 6.88. This result shows the noisiness of the data and the importance of pre-filtering.

Based on table 1 it can be seen that our default system, without post-ranking, could already achieve good performance. The additional post-ranking steps were most helpful for the setups with only 10M tokens in the training data. This indicates that giving less weight to redundant and not fluent sentences is especially important in the low resource setups. During the development we also performed an ablation study on the post-ranking methods. Using only the language model on top of pre-filtering and scoring gave 20.67 BLEU points while activating only the document similarity module we got 21.66 with SMT 10M. This shows that the latter method is more important because it removes more redundant data from the training set and makes space for sentence pairs that contain additional lexical information. On the other hand, language modeling causes lower performance increase because the rule-based pre-filtering step could already detect and remove some of the less fluent candidates. By combining the two techniques we could achieve the best performance on the newstest 2017 dataset. In contrast, post-ranking steps only helped for the iwslt2017 and Acquis datasets in the case of the 100M token setups. We conjecture that the down-weighting of candidates by these steps was too heavy which resulted in lower importance of these candidates comparing to candidates which are not even parallel. This issue could be overcome by better fine tuning of hyperparameters.

In table 2 we show the averaged results over all test sets of the best system of the official participants. Our systems performs better then the average in three out of four cases and scores below the best system by only 2.17 BLEU points on average. Our results are less competitive with NMT which is because we only used SMT during development. Our results show that competitive performance can be achieved without the use of any bilingual signal for the parallel corpus filtering task.

## 5 Conclusion

In this paper we introduced LMU Munich's submission to the WMT 2018 Parallel Corpus Filtering shared task. Such systems are especially useful in low resource setups, so we proposed a fully unsupervised system which is built on three modules: (i) we apply a pre-filtering step to remove noisy data (ii) we score sentences based on bilingual word embeddings and (iii) as a post-ranking step we penalize sentence pairs which are redundant or not fluent enough. We achieved good results with all setups which shows the competitiveness of our unsupervised system.

## Acknowledgments

We would like to thank Fabienne Braune, Yuliya Kalasouskaya and the anonymous reviewers for their valuable input. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement № 640550).

## References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58.

Fabienne Braune, Viktor Hangya, Tobias Eder, and Alexander Fraser. 2018. Evaluating bilingual word embeddings on the long tail. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 188–193.

Christian Buck and Philipp Koehn. 2016. Quick and reliable document alignment via tf/idf-weighted cosine distance. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 672–678.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word Translation Without Parallel Data. *CoRR*, abs/1710.04087.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The cmu-avenue french-english translation system. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 261–266.

Miquel Esplà-Gomis and Mikel Forcada. 2010. Combining content-based and url-based heuristics to harvest aligned bitexts from multilingual sites with bitextor. *The Prague Bulletin of Mathematical Linguistics*, pages 77–86.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *CoRR*, abs/1702.08734.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pages 116–121.

Tomoyuki Kajiwara and Mamoru Komachi. 2016. Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics on interactive poster and demonstration sessions*, pages 177–180.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 957–966.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.

Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, page 247.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. *arXiv preprint arXiv:1708.00712*.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950.

# Dual Conditional Cross-Entropy Filtering of Noisy Parallel Corpora

**Marcin Junczys-Dowmunt**
Microsoft
1 Microsoft Way
Redmond, WA 98121, USA

## Abstract

In this work we introduce dual conditional cross-entropy filtering for noisy parallel data. For each sentence pair of the noisy parallel corpus we compute cross-entropy scores according to two inverse translation models trained on clean data. We penalize divergent cross-entropies and weigh the penalty by the cross-entropy average of both models. Sorting or thresholding according to these scores results in better subsets of parallel data. We achieve higher BLEU scores with models trained on parallel data filtered only from Paracrawl than with models trained on clean WMT data. We further evaluate our method in the context of the WMT2018 shared task on parallel corpus filtering and achieve the overall highest ranking scores of the shared task, scoring top in three out of four subtasks.

## 1 Introduction

Recently, large web-crawled parallel corpora which are meant to rival non-public resources held by popular machine translation providers have been made publicly available to the research community in form of the Paracrawl corpus.[1] At the same time, it has been shown that neural translation models are far more sensitive to noisy parallel training data than phrase-based statistical machine translation methods (Khayrallah and Koehn, 2018; Belinkov and Bisk, 2017). This creates the need for data selection methods that can filter harmful sentence pairs from these large resources.

In this paper, we introduce dual conditional cross-entropy filtering, a simple but effective data selection method for noisy parallel corpora. We think of it as the missing adequacy component to the fluency aspects of cross-entropy difference filtering by Moore and Lewis (2010). Similar to Moore-Lewis filtering for monolingual data, we directly select samples that have the potential to improve perplexity (and in our case translation performance) of models trained with the filtered data.

This is different from Axelrod et al. (2011) who simply expand Moore and Lewis filtering to both sides of the parallel corpus. We use conditional probability distributions and enforce agreement between inverse translation directions.

In most cases, neural translation models are trained to minimize perplexity (or cross-entropy) on a training set. Our selection criterion includes the optimization criterion of neural machine translation which we approximate by using neural translation models pre-trained on clean seed data.

We evaluated our method in the context of the WMT2018 Shared Task on Parallel Corpus Filtering (Koehn et al., 2018) and submitted our best method to the task. Although we only optimized for one of the four subtasks of the shared task, our submission scored highest for three out of four subtasks and third for the fourth subtask; there were 48 submissions to each subtask in total.

## 2 WMT 2018 shared task on parallel corpus filtering

We quote the shared task description provided by the organizers on the task website[2] and add citations where appropriate: The organizers "provide a very noisy 1 billion word (English token count) German-English corpus crawled from the web as part of the Paracrawl project" and "ask participants to subselect sentence pairs that amount to (a) 100 million words, and (b) 10 million words. The quality of the resulting subsets is determined by the quality of a statistical machine translation — Moses, phrase-based (Koehn et al., 2007) — and a neural machine translation system — Mar-

[1] https://paracrawl.eu

[2] http://www.statmt.org/wmt18/parallel-corpus-filtering.html

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 888–895
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64105

ian (Junczys-Dowmunt et al., 2018) — trained on this data." The organizers note that the task is meant to address "the challenge of data quality and not domain-relatedness of the data for a particular use case." They discourage participants from subsampling the corpus for relevance to the news domain and announce that more emphasis will be put on undisclosed test sets rather than the WMT2018 test set.

Furthermore the organizers remark that "the provided raw parallel corpus is the outcome of a processing pipeline that aimed from high recall at the cost of precision, so it is very noisy. It exhibits noise of all kinds (wrong language in source and target, sentence pairs that are not translations of each other, bad language, incomplete or bad translations, etc.)" It is allowed to use the 2018 news translation task data for German-English (without the Paracrawl parallel corpus) to train components of our methods.

### 2.1 Sub-sampling based on submitted scores

Participants submit files with numerical scores, one score per line of the original unfiltered parallel corpus. A tool provided by the organizers takes as input the scores and the German and English corpus halves in form of raw text. Higher scores are better. The tool first determines at which best thresholds 10M and 100M words can be collected and next creates two data sets containing all sentences with scores above the two selected respective thresholds. Systems trained on these data sets are used for evaluation by the organizers (4 systems per submission) and for development purposes by task participants.

We focus on the 100M sub-task for neural machine translation systems as this is closest to our interests of finding as much relevant data as possible in large noisy parallel corpora. We only develop systems for this scenario.

### 2.2 Neural machine translation evaluation

As required by the shared task, we use Marian (Junczys-Dowmunt et al., 2018) to train our development systems. We choose hyper-parameters that favor quicker convergence during our own development phase. We follow the recommended settings quite closely in terms of model architecture, but change training settings. We switched off synchronous ADAM in favor of asynchronous ADAM, increased the evaluation frequency to once per 5000 updates and increased work-space size to 5000MB per GPU. We also set the initial learning-rate to 0.0003 instead of 0.0001 and used an inverse square-root decaying scheme for the learning rate (Vaswani et al., 2017) that started after 16,000 updates. We removed dropout of source and target words and decreased variational dropout from 0.2 to 0.1 (Gal and Ghahramani, 2016). With these settings, our models usually converged within 10 to 15 hours of training on four NVidia Titan Xp GPUs. Convergence was assumed if perplexity did not improve for 5 consecutive evaluation steps. We evaluated on the provided WMT2016 and WMT2017 test sets.

## 3 Scores and experiments

We produce a single score $f(x, y)$ per sentence pair $(x, y)$ as the product of partial scores $f_i(x, y)$:

$$f(x, y) = \prod_i f_i(x, y). \tag{1}$$

Partial scores take values between 0 and 1, as does the total score $f$. Partial scores that might generate values outside that range are clipped. We assume that sentence pairs with a score of 0 are excluded from the training data.[3]

In this section, we describe the scores explored in this work and present results on the development data.

### 3.1 Experimental baselines

Following the training recipe in Section 2.2, we first trained a model ("WMT18-full" in Table 2) on the admissible parallel WMT18 data for German-English (excluding Paracrawl). This model is only used for the computation of reference BLEU scores.

Next, we trained a German-English model on randomly scored Paracrawl data only ("random" in Table 2). The random scores – uniformly sampled values between 0 and 1 – were used to select representative data consisting of 100M words from unprocessed Paracrawl while using the threshold-based selection tool provided by the shared task organizers. Results for WMT16 and WMT17 test sets for both systems are shown in Table 2. The Paracrawl-trained systems (random) has dramatically worse BLEU scores than the WMT18-trained system. Upon manual inspection, we see many

[3] This is only guaranteed by the selection algorithm of the shared task if more than 100M words appear in sentence pairs scored with non-zero scores. However, we did not encounter situations where we got close or below that boundary.

| Model | Description |
|---|---|
| $W_{\text{en}}$ | RNN language model trained on 1M sentences from English WMT monolingual news data 2015-2017 |
| $P_{\text{en}}$ | RNN language model trained on 1M sentences from target (English) side of Paracrawl |
| $W_{\text{de}\rightarrow\text{en}}$ | German-English translation model trained on WMT parallel data |
| $W_{\text{en}\rightarrow\text{de}}$ | English-German translation model trained on WMT parallel data |
| $W_{\text{de}\leftrightarrow\text{en}}$ | Translation model trained on union of German-English and English-German WMT parallel data |

Table 1: Helper models trained for various scorers. All models are neural models, we do not use n-gram or phrase-based models. WMT parallel data excludes Paracrawl data.

untranslated and partially copied sentences in the case of the randomly-selected Paracrawl system.

### 3.2 Language identification

We noticed that the provided sentence pairs do not seem to have been subjected to language identification and simply used the Python `langid` package to assign a language code to each sentence in a sentence pair. We did not restrict the inventory of languages beforehand as we wanted the tool to propose a language if that language wins against all other candidates. We only accepted sentence pairs where both elements of a pair had been assigned the desired languages (German for source, English for target). The result is our first non-trivial score:

$$\text{lang}(x, l) = \begin{cases} 1 & \text{if } \textsc{LangID}(x) = l \\ 0 & \text{otherwise} \end{cases}$$

$$\text{de-en}(x, y) = \text{lang}(x, \text{“de”}) \cdot \text{lang}(y, \text{“en”}) \quad (2)$$

This is a very harsh but also very effective filter that removes nearly 70% of the parallel sentence candidates. As a beneficial side-effect of language identification many language-ambiguous fragments which contain only little textual information are discarded, e.g. sentences with lots of numbers, punctuation marks or other non-letter characters. The identification tool gets confused by the non-textual content and selects a random language.

We combined the $\text{de-en}(\text{x}, \text{y})$ filter with the random scores and trained a corresponding system (de-en·random). As we see in Table 2, this strongly improved the results on both dev sets. When reviewing the translated development sets, we did not see any copied/untranslated sentences in the output.

### 3.3 Dual conditional cross-entropy filtering

The scoring method introduced in this section is our main contribution. While inspired by cross-entropy difference filtering for monolingual data (Moore and Lewis, 2010), our method does not aim for monolingual domain-selection effects. Instead we try to model a bilingual adequacy score.

Moore and Lewis (see next section) quantify the directed disagreement (signed difference) of similar distributions (two language models over the same language) trained on dissimilar data (different monolingual corpora). A stronger degree of separation between the two models indicates more interesting data.

In contrast, we try to find maximal symmetric agreement (minimal absolute difference) of dissimilar distributions (two translation models over inverse translation directions) trained on the same data (same parallel corpus). Concretely, for a sentence pair $(x, y)$ we calculate a score:

$$\begin{aligned} &|H_A(y|x) - H_B(x|y)| \\ &+\frac{1}{2}\left(H_A(y|x) + H_B(x|y)\right) \end{aligned} \quad (3)$$

where $A$ and $B$ are translation models trained on the same data but in inverse directions, and $H_M(\cdot|\cdot)$ is the word-normalized conditional cross-entropy of the probability distribution $P_M(\cdot|\cdot)$ for a model $M$:

$$\begin{aligned} H_M(y|x) &= -\frac{1}{|y|}\log P_M(y|x) \\ &= -\frac{1}{|y|}\sum_{t=1}^{|y|}\log P_M(y_t|y_{<t}, x). \end{aligned}$$

The score (denoted as dual conditional cross-entropy) has two components with different functions: the absolute difference $|H_A(y|x) - H_B(x|y)|$ measures the agreement between the two conditional probability distributions, assuming that (word-normalized) translation probabilities of sentence pairs in both directions should be roughly equal. We want disagreement to be low, hence this value should be close to 0.

However, a translation pair that is judged to be equally improbable by both models will also have a low disagreement score. Therefore we weight the agreement score by the average word-normalized cross-entropy from both models. Improbable sentence pairs will have higher average cross-entropy values.

This score is also quite similar to the dual learning training criterion from He et al. (2016) and Hassan et al. (2018). The dual learning criterion is formulated in terms of joint probabilities, later decomposed into translation model and language model probabilities. In practice, the influence of the language models is strongly scaled down which results in a form more similar to our score.

While Moore and Lewis filtering requires an in-domain data set and a non-domain-specific data set to create helper models, we require a clean, relative high-quality parallel corpus to train the two dual translation models. We sample 1M sentences from WMT parallel data excluding Paracrawl and train Nematus-style translation models $W_{\text{de}\to\text{en}}$ and $W_{\text{en}\to\text{de}}$ (see Table 1).

Formula (3) produces only positive values with 0 being the best possible score. We turn it into a partial score with values between 0 and 1 (1 being best) by negating and exponentiating, setting $A = W_{\text{de}\to\text{en}}$ and $B = W_{\text{en}\to\text{de}}$:

$$\begin{aligned}\text{adq}(x,y) = \exp(-(&|H_A(y|x) - H_B(x|y)| \\ &+ \frac{1}{2}(H_A(y|x) + H_B(x|y))).\end{aligned}$$

Combining the adq filter with the de-en filter results in a promising NMT system (de-en · adq in Table 2) trained on Paracrawl alone that beats the BLEU scores of the pure-WMT baseline.

We further evaluated three ablative systems:

- we omitted the language id filter (no de-en) which resulted in a system worse than randomly selected. This is not too surprising as we would expect many identical strings to be selected as highly adequate;
- we dropped the absolute difference from formula (3) which decreased BLEU by about 1 point;
- we removed the weighting by the averaged cross-entropies from formula (3), loosing about 3 BLEU points.

This seems to indicate that the two components of the dual conditional cross-entropy filter are indeed useful and that we have a practical scoring method for parallel data.

### 3.4 Cross-entropy difference filtering

When inspecting the training data generated with the above methods we saw many fragments that looked like noisy or not particularly useful data. This included concatenated lists of dates, series of punctuation marks or simply not well-formed text. Due to the adequacy filtering, the noise was at least adequate, i.e. similar or identical on both sides and mostly correctly translated if applicable. The language filter had made sure that only few fully identical pairs of fragments had remained.

However, we preferred to have a training corpus that also looked like clean data. To achieve this we treated cross-entropy filtering proposed by Moore and Lewis (2010) as another score. Cross-entropy filtering or Moore-Lewis filtering uses the quantity

$$H_I(x) - H_N(x) \tag{4}$$

where $I$ is an in-domain model, $N$ is a non-domain-specific model and $H_M$ is the word-normalized cross-entropy of a probability distribution $P_M$ defined by a model $M$:

$$\begin{aligned}H_M(x) &= -\frac{1}{|x|}\log P_M(x) \\ &= -\frac{1}{|x|}\sum_{t=1}^{|x|}\log P_M(x_t|x_{<t}).\end{aligned}$$

Sentences scored with this method and selected when their score is below a chosen threshold are likely to be more in-domain according to model $I$ and less similar to data used to train $N$ than sentences above that threshold.

We chose WMT English news data from the years 2015-2017 as our in-domain, clean language model data and sampled 1M sentences to train model $I = W_{\text{en}}$. We sampled 1M sentences from Paracrawl without any previously applied filtering to produce $N = P_{\text{en}}$. The shared task organizers encourage submitting teams to not optimize for a specific domain, but it has been our experience that news data is quite general and clean data beats noisy data on many domains.

To create a partial score for which the best sentence pairs produce a 1 and the worst at 0, we apply a number of transformations. First, we negate and exponentiate cross-entropy difference arriving at a quotient of perplexities of the target sentence $y$ ($x$ is ignored):

$$\begin{aligned} \mathrm{dom}'(x, y) &= \exp(-(H_I(y) - H_N(y))) \\ &= \frac{\mathrm{PP}_N(y)}{\mathrm{PP}_I(y)}. \end{aligned}$$

This score has the nice intuitive interpretation of how many times sentence $y$ is less perplexing to the in-domain model $W_{\mathrm{en}}$ than to the out-of-domain model $P_{\mathrm{en}}$.

We further clip the maximum value of the score to 1 (the minimum value is already 0) as:

$$\mathrm{dom}(x, y) = \max(\mathrm{dom}'(x, y), 1). \quad (5)$$

This seems counterintuitive at first, but is done to avoid that a high monolingual in-domain score strongly overrides bilingual adequacy; we are fine with low in-domain scores penalizing sentence pairs. This is a precision-recall trade-off for adequacy and we prefer precision.

Finally, we also propose a cut-off value $c$ as a parameter:

$$\mathrm{cut}(x, c) = \begin{cases} x & \text{if } x \geq c \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{dom}_c(x, y) = \mathrm{cut}(\mathrm{dom}(x, y), c). \quad (6)$$

Parameter $c$ can be used to completely eliminate sentence pairs, regardless of other scores, if $y$ is less than $c$ times more perplexing to the out-of-domain model than to the in-domain model, or inversely $1/c$ times more perplexing to the in-domain model than the out-of-domain model. This seems useful if we want a hard noise-filter similar to the language-id filter described above.

We used the domain filter only in combination with the previously introduce filters. In Table 2, we can observe that any variant leads to small improvements of the model over variants without the dom filters. This is expected as we optimized for WMT news development sets. We experimented with three cut-off values: 0.00 (no cut-off), 0.25 and 0.50, reaching the highest BLEU scores for a cut-off value $c = 0.25$. This best result (bold in Table 2) was submitted to the shared task organizers as our only submission.

Future work should consider bilingual cross-entropy difference filtering as proposed by Axelrod et al. (2011) where both sides of the corpus undergo the selection process or experiment with conditional probability distributions (translation models) for domain filtering.

| Filter | test16 | test17 |
|---|---|---|
| WMT18-full | 33.9 | 29.0 |
| random | 16.2 | 14.1 |
| de-en·random | 26.6 | 23.3 |
| de-en·adq | 35.1 | 30.2 |
| - no de-en | 15.4 | 12.7 |
| - no absolute difference | 33.8 | 29.3 |
| - no CE weighting | 31.7 | 27.4 |
| de-en·adq·dom$_{0.00}$ | 35.5 | 30.5 |
| **de-en·adq·dom$_{0.25}$** | **36.0** | **31.0** |
| de-en·adq·dom$_{0.50}$ | 35.4 | 30.6 |
| de-en·sim | 34.5 | 29.6 |
| de-en·sim·dom$_{0.25}$ | 35.5 | 30.6 |
| de-en·adq·sim·dom$_{0.25}$ | 35.5 | 30.7 |

Table 2: Results on development data. We only train neural models for the 100M sub-task. We did not optimize for any of the other three sub-tasks.

### 3.5 Cosine similarity of sentence embeddings

We further experimented with sentence embedding similarity to contrast this method with our cross-entropy based approach. Recently, Hassan et al. (2018) and Schwenk (2018) used cosine similarities of sentence embeddings in a common multi-lingual space to select translation pairs for neural machine translation. Both these approaches rely on creating a multi-lingual translation model across all available translation directions and then using the accumulated encoder representations (after summing or max-pooling contextual word-level embeddings across the time dimension) of sentences in a pair to compute similarity scores.

Following Hassan et al. (2018), we train a new multi-lingual translation model on WMT18 parallel data (excluding Paracrawl) by joining German-English and English-German training data into a mixed-direction training set (see model $W_{\mathrm{de}\leftrightarrow\mathrm{en}}$ in Table 1). For a given sentence $x$, we create its sentence embedding vector $\mathbf{s}_x$ according to translation model $W_{\mathrm{de}\leftrightarrow\mathrm{en}}$ by collecting encoder representation vectors $\mathbf{h}_1$ to $\mathbf{h}_{|x|}$

$$\mathbf{h}_{1:|x|} = \mathrm{Encoder}_{W_{\mathrm{de}\leftrightarrow\mathrm{en}}}(x) \quad (7)$$

which are then averaged to form a single vector

| System | Avg-BLEU |
|---|---|
| RWTH Neural Redund. | 24.58 |
| RWTH Neural Indep. | 24.53 |
| **Our submission** | **24.45** |
| AliMT Mix | 24.11 |
| AliMT Mix-div | 24.11 |

(a) SMT 10M

| System | Avg-BLEU |
|---|---|
| **Our submission** | **26.50** |
| AliMT Mix | 26.44 |
| AliMT Mix-div | 26.42 |
| Prompsit Active | 26.41 |
| NRC yisi-bicov | 26.40 |

(b) SMT 100M

| System | Avg-BLEU |
|---|---|
| **Our submission** | **28.62** |
| RWTH Neural Redund. | 28.01 |
| RWTH Neural Indep. | 28.00 |
| Speechmatics best | 27.97 |
| Speechmatics prime | 27.88 |

(c) NMT 10M

| System | Avg-BLEU |
|---|---|
| **Our submission** | **32.05** |
| AliMT Mix | 31.93 |
| AliMT Mix-div | 31.92 |
| NRC yisi-bicov | 31.88 |
| NRC yisi | 31.76 |

(d) NMT 100M

| System | Avg-BLEU |
|---|---|
| **Our submission** | **111.63** |
| RWTH Neural Redundancy | 110.09 |
| AliMT Mix | 110.07 |
| AliMT Mix-div | 110.05 |
| RWTH Neural Independent | 109.91 |

(e) Sum of all sub-tasks

Table 3: Top-5 out of 48 submissions for each of the four sub-tasks and total sum

representation

$$\mathbf{s}_x = \frac{1}{|x|} \sum_{t=1}^{|x|} \mathbf{h}_t. \quad (8)$$

For a given sentence pair $(x, y)$ we compute the cosine similarity of $\mathbf{s}_x$ and $\mathbf{s}_y$ as

$$\mathrm{sim}(x, y) = \cos(\measuredangle \mathbf{s}_x \mathbf{s}_y) = \frac{\mathbf{s}_x \cdot \mathbf{s}_y}{|\mathbf{s}_x||\mathbf{s}_y|}. \quad (9)$$

Since the model has seen both languages, English and German, as source data it can produce useful sentence representations of both sentences in a translation pair. Unlike Hassan et al. (2018), we did not define a cut-off value for the similarity score as the threshold-based selection method of shared-task tool computes its own cut-off thresholds.

We ran two experiments with the similarity based scores, evaluating configurations de-en·sim and de-en·adq·sim·dom$_{0.25}$. The first one corresponds to de-en·adq and we compare the effectivness of the adq and sim filters after the application of the language-id-based filter de-en. We see in Table 2 that while de-en·sim leads to improvements over the language-filtered randomly selected Paracrawl data, it is significantly worse than de-en·adq on both development sets. Interestingly, even when combined with our best scoring scheme (de-en·adq·dom$_{0.25}$) resulting in de-en·adq·sim·dom$_{0.25}$ we see a slight degradation. Based on these results, we do not use the similarity scores for our submission.

In future experiments we want to use the multilingual model $W_{\mathrm{de}\leftrightarrow\mathrm{en}}$ instead of the two models $W_{\mathrm{en}\rightarrow\mathrm{de}}$ and $W_{\mathrm{de}\rightarrow\mathrm{en}}$ for our dual conditional cross-entropy method from Section 3.3. A multilingual model does not only have a common encoder, but also a common probability distribution for both languages which might lead to better agreement of the conditional cross-entropies.

## 4 Shared task results

As mentioned before, we submitted only our single-best set of scores $\mathrm{de\text{-}en} \cdot \mathrm{adq} \cdot \mathrm{dom}_{0.25}$ to the shared task. The shared task organizers trained four systems with each set of submitted scores, two Moses SMT (Koehn et al., 2007) systems on the best 10M and 100M words corpora and two neural Marian NMT systems on the same sets.

Based on the spread-sheet made available by the organizers, 48 sets of scores where submitted. Each set of scores was evaluated using the four mentioned models on 6 different test sets (newstest 2018, iwslt 2017, Acquis, EMEA, Global Voices, KDE). This required the organizers to train nearly 200 separate models; an effort that should be applauded.

It seems that systems are ranked by their average score across these test sets and sub-tasks. In Table 3 we selected the top-5 system across each sub-task for the purpose of this paper. The shared task overview will likely include a more thorough analysis. We place highest out of 48 submissions in three out of four tasks (SMT 100M, NMT 10M and NMT 100M) and third out of 48 for sub-task SMT 10M. The systems are packed quite closely, but the overall total across all four tasks shows, that we accumulate a slightly larger margin over the next best systems while the next four systems barely differ. This result is better than we expected as we only optimized for the NMT 100M task.

For more details on the evaluation process and conclusions see the shared task overview paper Koehn et al. (2018).

## 5 Future work and discussion

We introduced dual conditional cross-entropy filtering for noisy parallel data and combined this filtering with multiple other noise filtering methods. Our submission to the WMT 2018 shared task on parallel corpus filtering achieved the highest overall rank and scored best in three out of four subtasks while scoring third in the fourth subtask. Each subtask had 48 participants.

We believe this positive effect is rooted in the idea of directly asking a model that is very similar to the to-be-trained model which data it prefers (weighting by cross-entropy) while also constraining its answer with the introduced disagreement penalty. Our selection criterion is also very close to the optimization criterion used during NMT training, especially the dual learning training criterion. Other methods, for instance the evaluated similarity-based methods, do not have this direct connection to the training process.

Future work should concentrate on further formalizing this method. We should analyze the connection to the dual learning training criterion on experiments whether models that were trained with this criterion are also better candidates for sentences scoring. Furthermore, the models we used for scoring were trained on small subsamples of clean data, we should investigate if stronger translation and language models are better discriminators.

## References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 355–362. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. CoRR, abs/1711.02173.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In Advances in neural information processing systems, pages 1019–1027.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. CoRR, abs/1803.05567.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 820–828. Curran Associates, Inc.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In Proceedings of ACL 2018, System Demonstrations, pages 116–121. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

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. 2007. Moses: Open source toolkit for statistical machine translation. In ACL. The Association for Computer Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers, Brussels, Belgium. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In Proceedings of the ACL 2010 Conference Short Papers, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Holger Schwenk. 2018. Filtering and mining parallel data in a joint multilingual space. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 228–234. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc.

# The JHU Parallel Corpus Filtering Systems for WMT 2018

**Huda Khayrallah** **Hainan Xu** **Philipp Koehn**
Center for Language and Speech Processing
Johns Hopkins University
{huda, hxu31, phi}@jhu.edu

## Abstract

This work describes our submission to the WMT18 Parallel Corpus Filtering shared task. We use a slightly modified version of the Zipporah Corpus Filtering toolkit (Xu and Koehn, 2017), which computes an adequacy score and a fluency score on a sentence pair, and use a weighted sum of the scores as the selection criteria. This work differs from Zipporah in that we experiment with using the noisy corpus to be filtered to compute the combination weights, and thus avoids generating synthetic data as in standard Zipporah.

## 1 Introduction

Todays machine translation systems require large amounts of training data in form of sentences paired with their translation, which are often compiled from online sources. This has not changed fundamentally with the move from statistical machine translation to neural machine translation, also we observed that neural models require more training data (Koehn and Knowles, 2017) and are more sensitive to noise (Khayrallah and Koehn, 2018). Thus both the acquisition of more training data such as indiscriminate web crawling and corpus filtering will have large impact on the quality of state-of-the-art machine translation systems.

The JHU submission to the WMT18 Parallel Corpus Filtering shared task uses a modified version of the Zipporah Corpus Filtering toolkit (Xu and Koehn, 2017). For a sentence pair, Zipporah uses a bag-of-words model to generate an adequacy score, and an n-gram language model to generate fluency score. The two scores are combined based on weights trained in order to separate clean data from noisy data. The original version of Zipporah generates artificial noisy training data to train such classifier, in this submission we also treat the Paracrawl corpus as the negative examples.

## 2 Related Work

Zipporah builds upon prior work in data cleaning and data selection.

For data selection, work has focused on selecting a subset of data based on domain-matching. Moore and Lewis (2010) computed cross-entropy between in-domain and out-of-domain language models to select data for training domain-relevant language models. XenC (Rousseau, 2013), an open-source tool, also selects data based on cross-entropy scores on language models. Axelrod et al. (2015) utilized part-of-speech tags and used a class-based n-gram language model for selecting in-domain data and Duh et al. (2013) used a neural network based language model trained on a small in-domain corpus to select from a larger mixed-domain data pool. Lü et al. (2007) redistributed different weights for sentence pairs/predefined sub-models. Shah and Specia (2014) described experiments on quality estimation which, given a source sentence, select the best translation among several options.

For data cleaning, work has focused on removing noisy data. Taghipour et al. (2011) proposed an outlier detection algorithm which leads to an improved translation quality when trimming a small portion of data. Cui et al. (2013) used a graph-based random walk algorithm to do bilingual data cleaning. BiTextor (Esplá-Gomis and Forcada, 2009) utilizes sentence alignment scores and source URL information to filter out bad URL pairs and selects good sentence pairs. Similar to this work, the qe-clean system (Denkowski et al., 2012; Dyer et al., 2010; Heafield, 2011) uses word alignments and language models to select sentence pairs that are likely to be good translations of one another.

We focus on data cleaning for all purposes, as opposed to data selection for a given domain. We

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 896–899
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64106

aim to create a corpus of generally valid translations, which could then be filtered to adapt to a particular domain.

# 3 Zipporah

We use a slightly modified version of the Zipporah Corpus Filtering toolkit (Xu and Koehn, 2017). Zipporah works as follows: it first maps all sentence pairs into the proposed feature space, and then trains a simple logistic regression model to separate known good data and bad data. Once the model is trained, it is used to score sentence pairs in the noisy data pool.

Zipporah uses two features inspired by *adequacy* and *fluency*. The adequacy feature uses bag-of-words translation scores, and the fluency feature uses n-gram language model scores.

## 3.1 Adequacy Score

Zipporah generates probabilistic dictionaries from an aligned corpus, and uses them to generate bag of words translation scores for each sentence. This is done in both directions.

Given a sentence pair $(s_f, s_e)$ in the noisy data pool, we represent the two sentence as two sparse word-frequency vectors $v_f$ and $v_e$. For example for any French word $w_f$, we have $v_f[w_f] = \frac{c(w_f, s_f)}{l(s_f)}$, where $c(w_f, s_f)$ is the number of occurrences of $w_f$ in $s_f$ and $l(s_f)$ is the length of $s_f$. We do the same for $v_e$. Then we "translate" $v_f$ into $v'_e$, based on the probabilistic f2e dictionary, where

$$v'_e[w_e] = \sum_{w_f} v_f[w_f] p(w_e|w_f)$$

For a French word $w$ that does not appear in the dictionary, we keep it as it is in the translated vector, i.e. assume there is an entry of $(w, w, 1.0)$ in the dictionary. We compute the cross-entropy between $v_e$ and $v'_e$,

$$\text{xent}(v_e, v'_e) = \sum_{w_e} v_e[w_e] \log \frac{1}{v'_e[w_e] + c} \qquad (1)$$

where $c$ is a smoothing constant to prevent the denominator from being zero, which we set $c = 0.0001$ for all experiments.

We perform similar procedures for English-to-French, and compute $\text{xent}(v_f, v'_f)$. We define the adequacy score as the sum of the two:

$$\text{adequacy}(s_f, s_e) = \text{xent}(v_e, v'_e) + \text{xent}(v_f, v'_f)$$

## 3.2 Fluency Score

Zipporah trains two 5-gram language models with a clean French and English corpus, and then for each sentence pair $(s_g, s_e)$ scores each sentence with the corresponding model, $\mathcal{F}_{\text{ngram}}(s_g)$ and $\mathcal{F}_{\text{ngram}}(s_e)$, each computed as the ratio between the sentence negative log-likelihood and the sentence length. We define the fluency score as the sum of the two:

$$\text{fluency}(s_G, s_e) = \mathcal{F}_{\text{ngram}}(s_G) + \mathcal{F}_{\text{ngram}}(s_e)$$

## 3.3 Classifier

We train a binary classifier to separate a clean corpus from noisy corpora, based on the 2 features proposed. Higher orders of the features are used in order to achieve a non-linear decision boundary. We implement this using the logistic regression model from scikit-learn (Pedregosa et al., 2011), and use the features in the form of $(x^8, y^8)$.

## 3.4 Training Data

We use clean WMT training data as the examples of clean text. The original version of Zipporah creates synthetic negative training examples by shuffling the clean data set, both at the corpus and sentence levels in order to generate inadequate and non-fluent text.

Since much of the raw Paracrawl data is noisy (Khayrallah and Koehn, 2018), we also train a version where we simply use the portion of Paracrawl released for the shared task as the negative examples to train our classifier, without generating synthetic noisy data. We experiment with using both the full portion of Paracrawl and a $10,000$ line subset.

# 4 Results

We include the results of running the three versions of Zipporah in Table 1. The final column is the average score across the 6 test sets.

- Zipporah-synthetic denotes the system with synthetic negative examples as in the original version of Zipporah.
- Zipporah-paracrawl denotes the system trained with the Paracrawl as the negative examples.
- Zipporah-paracrawl-10000 denotes the system trained with a 10000 sentence subset of Paracrawl.

**Statistical machine translation (SMT) scores, 10 million words**

| | dev | test | | | | | | |
|---|---:|---:|---:|---:|---:|---:|---:|---:|
| System Name | Newstest2017 | Newstest2018 | IWSLT | Acquis | EMEA | GV | KDE | **avg** |
| zipporah-synthetic | 21.77 | 26.75 | 20.78 | 19.40 | 25.07 | 20.70 | 24.45 | **22.85** |
| zipporah-paracrawl-10000 | 20.24 | 26.31 | 20.21 | 19.88 | 24.69 | 20.28 | 24.30 | **22.61** |
| zipporah-paracrawl | 20.18 | 26.26 | 20.36 | 19.33 | 24.76 | 20.37 | 24.32 | **22.57** |
| best shared task submission | 23.14 | 29.59 | 21.76 | 21.45 | 28.12 | 22.63 | 23.93 | **24.58** |

**Statistical machine translation scores (SMT), 100 million words**

| | dev | test | | | | | | |
|---|---:|---:|---:|---:|---:|---:|---:|---:|
| System Name | Newstest2017 | Newstest2018 | IWSLT | Acquis | EMEA | GV | KDE | **avg** |
| zipporah-synthetic | 24.93 | 30.32 | 22.79 | 22.42 | 30.13 | 23.40 | 26.57 | **25.94** |
| zipporah-paracrawl-10000 | 24.85 | 30.19 | 22.61 | 22.12 | 29.92 | 23.35 | 26.42 | **25.77** |
| zipporah-paracrawl | 24.81 | 30.35 | 22.63 | 22.13 | 30.12 | 23.26 | 26.52 | **25.84** |
| best shared task submission | 25.80 | 31.35 | 23.17 | 22.29 | 31.45 | 23.88 | 26.87 | **26.50** |

**Neural machine translation scores (NMT), 10 million words**

| | dev | test | | | | | | |
|---|---:|---:|---:|---:|---:|---:|---:|---:|
| System Name | Newstest2017 | Newstest2018 | IWSLT | Acquis | EMEA | GV | KDE | **avg** |
| zipporah-synthetic | 26.13 | 32.22 | 23.89 | 22.73 | 26.95 | 24.26 | 24.94 | **25.83** |
| zipporah-paracrawl-10000 | 25.21 | 31.44 | 23.13 | 22.82 | 26.31 | 24.02 | 24.32 | **25.34** |
| zipporah-paracrawl | 25.20 | 31.31 | 23.14 | 22.51 | 26.56 | 24.38 | 24.53 | **25.41** |
| best shared task submission | 28.49 | 35.67 | 25.10 | 23.69 | 32.72 | 26.72 | 27.81 | **28.62** |

**Neural machine translation scores (NMT), 100 million words**

| | dev | test | | | | | | |
|---|---:|---:|---:|---:|---:|---:|---:|---:|
| System Name | Newstest2017 | Newstest2018 | IWSLT | Acquis | EMEA | GV | KDE | **avg** |
| zipporah-synthetic | 29.59 | 36.42 | 24.61 | 27.60 | 35.47 | 27.50 | 29.57 | **30.20** |
| zipporah-paracrawl-10000 | 29.56 | 36.75 | 24.24 | 27.57 | 35.36 | 27.70 | 29.32 | **30.16** |
| zipporah-paracrawl | 29.13 | 36.43 | 23.25 | 27.26 | 35.06 | 27.32 | 29.20 | **29.75** |
| best shared task submission | 32.41 | 39.85 | 27.43 | 28.31 | 36.70 | 29.26 | 30.79 | **32.06** |

Table 1: Results of our Zipporah variants, compared to the submission with the best average test score.

In general, our systems lag behind the top performing systems by about 3 BLEU on the average of the six test sets. The different Zipporah systems perform similarly, with a slight edge to the original version with synthetic parallel data. This indicates that a subset can be used for faster training of Zipporah.

Zipporah does not require building an initial NMT system to score the data, as required by some of the top performing systems. Zipporah also has a very fast run time, the most expensive part being the language model scoring.

Our submissions are more competitive in the SMT experiments, and lag behind the top performing system system by less than a BLEU point (averaged across the test sets) for SMT systems trained on 100 million sentences. This may be due to the fact that Zipporah's adequacy and fluency scores directly track the translation and language model components of SMT.

## 5 Conclusion

Our submission to the WMT 2018 shared task on parallel corpus filtering was based on our Zipoorah toolkit. We varied methods to generate negative samples for the classifier to detect noisy sentence pairs, with similar results for synthetic noise, the full raw corpus to be filtered, and a subset of it.

We note that our method is quite simple and fast, using only n-gram language model and bag-of-words translation model features.

## Acknowledgments

This work was in part supported by the IARPA MATERIAL project and a Google Faculty Research Award.

## References

Amittai Axelrod, Yogarshi Vyas, Marianna Martindale, Marine Carpuat, and Johns Hopkins. 2015. Class-based n-gram language difference models for data selection. In *IWSLT (International Workshop on Spoken Language Translation)*.

Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li, and Ming Zhou. 2013. Bilingual data cleaning for smt using graph-based random walk. In *ACL (2)*, pages 340–345.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The cmu-avenue french-english translation system. In *Proceedings of the NAACL 2012 Workshop on Statistical Machine Translation*.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *ACL (2)*, pages 678–683.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Miquel Esplá-Gomis and M Forcada. 2009. Bitextor, a free/open-source software to harvest translation memories from multilingual websites. *Proceedings of MT Summit XII, Ottawa, Canada. Association for Machine Translation in the Americas*.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Yajuan Lü, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *EMNLP-CoNLL*, volume 34, pages 3–350.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224. Association for Computational Linguistics.

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. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Anthony Rousseau. 2013. Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, 100:73–82.

Kashif Shah and Lucia Specia. 2014. Quality estimation for translation selection. In *Proceedings of the 17th Annual Conference of the European Association for Machine Translation, Dubrovnik, Croatia*.

Kaveh Taghipour, Shahram Khadivi, and Jia Xu. 2011. Parallel corpus refinement as an outlier detection algorithm. *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 414–421.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950. Association for Computational Linguistics.

# Measuring sentence parallelism using Mahalanobis distances: The NRC unsupervised submissions to the WMT18 Parallel Corpus Filtering shared task

**Patrick Littell, Samuel Larkin, Darlene Stewart,**
**Michel Simard, Cyril Goutte, Chi-kiu Lo**
National Research Council of Canada
1200 Montreal Road, Ottawa ON, K1A 0R6
`Firstname.Lastname@cnrc-nrc.gc.ca`

## Abstract

The WMT18 shared task on parallel corpus filtering (Koehn et al., 2018b) challenged teams to score sentence pairs from a large high-recall, low-precision web-scraped parallel corpus (Koehn et al., 2018a). Participants could use existing sample corpora (e.g. past WMT data) as a supervisory signal to learn what a "clean" corpus looks like. However, in lower-resource situations it often happens that the target corpus of the language is the *only* sample of parallel text in that language. We therefore made several unsupervised entries, setting ourselves an additional constraint that we not utilize the additional clean parallel corpora. One such entry fairly consistently scored in the top ten systems in the 100M-word conditions, and for one task—translating the European Medicines Agency corpus (Tiedemann, 2009)—scored among the best systems even in the 10M-word conditions.

## 1 Introduction and motivation

The WMT18 shared task on parallel corpus filtering assumes (but does not require) a supervised learning approach. Given

1. a set of "clean" German-English parallel corpora including past WMT data, Europarl (Koehn, 2005), etc., and
2. a large, potentially "dirty" corpus (i.e., one that may contain non-parallel data, non-linguistic data, etc.) scraped from the internet (Koehn et al., 2018a),

can one identify which sentences from (2) are clean? Supervised learning is an obvious approach in well-resourced languages like German and English, in which there exist well-cleaned parallel corpora across various domains.

However, in much lower-resourced languages, we generally do not have *multiple* parallel corpora in a given language pair to assess the quality of the corpus at hand; the corpus to be evaluated is often the only one available.[1] If we want to assess the quality of *one* corpus, we cannot rely on a supervisory signal derived from additional, cleaner corpora. We therefore do not utilize the additional parallel corpora (except as additional sources of monolingual data).

The systems described in this paper were inspired instead by anomaly detection approaches: can we instead attempt to identify sentence pairs that are, in some way, "strange" for this dataset? Considering each sentence pair as a draw from a distribution of high-dimensional vectors, we define an anomalous sentence pair as one whose draw was improbable compared to the probability of drawing its component sentences independently. The resulting measure, conceptually similar to pointwise mutual information albeit couched in terms of Mahalanobis distances rather than actual probabilities, is detailed in §3.

A submission based primarily on this one measurement (with some pre- and post-processing to avoid duplicate and near-duplicate sentences) performed consistently above the median in the 100M-word conditions, and for a few tasks (particularly EMEA translation) was among the top systems even for the 10M-word conditions. It was also the #2 system in one of the dev conditions (WMT newstest2017, NMT trained on 100M words), which is surprising given that it could not have overfit to the development set; it did not utilize the WMT17 development set in any way.

## 2 Overall architecture

The highest-ranked submission of our unsupervised submissions, `NRC-seve-bicov`,

[1] We are thinking in particular of the English-Inuktitut translation pair, which is a long-standing research interest of NRC (e.g. Martin et al., 2003).

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 900–907
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64107

shares the same general skeleton as NRC's highest-ranked supervised submission, NRC-yisi-bicov (Lo et al., 2018); it differs primarily in the parallelism estimation component (§2.3).

### 2.1 Training sentence embeddings

We began by training monolingual sentence embeddings using sent2vec (Pagliardini et al., 2018), on all available monolingual data. This included the monolingual data available in the "clean" parallel training data. That is to say, we did not completely throw out the clean parallel data for this task, we simply used it as two unaligned monolingual corpora.

We trained sentence vectors of 10, 50, 100, 300, and 700 dimensions; our final submissions used the 300-dimensional vectors as a compromise between accuracy (lower-dimensional vectors had lower accuracy during sanity-checking) and efficiency (higher-dimensional vectors ended up exceeding our memory capacity in downstream components).

In a system such as this, which is looking for "strange" sentence pairs, training on additional monolingual data beyond the target corpus carries some risks. If the additional monolingual data were to have very different domain characteristics (say, mostly religious text in the first language and mostly medical text in the second), then the two vector spaces could encode different types of sentence as "normal". On the other hand, not using additional monolingual data carries its own risks; monolingual data that *is* domain-balanced could help to mitigate domain mismatches in the target parallel data (say, newswire text erroneously misaligned to sequences of dates).

### 2.2 Pre-filtering

Although the input data had already been deduplicated by the shared task organizers, we did an additional de-duplication step in which email addresses and URLs were replaced with a placeholder token and numbers were removed, before deciding which sentences were duplicates. We had noticed that large amounts of data consisted of short sentences that were largely numbers (for example, long lists of dates). Although these sentences were indeed unique, we noticed that several of our parallelism measurements ended up preferring such sentences to such an extent that the resulting MT training sets were disproportionately dates, and performed comparatively poorly when tasked with training full sentences. To mitigate this, we ran an additional de-duplication step on the English side in which two sentences that differ only in numbers (e.g., "14 May 2017" and "19 May 1996") were considered duplicates.

Without numerical de-duplication, we believe the parallelism estimation step in §2.3 would have had too much of a bias towards short numerical sentences. It is, after all, essentially just looking for sentence pairs that it considers likely given the distribution of sentence pairs in the target corpus; if the corpus has a large number of short numerical sentences (and it appears to), the measurement will come to prefer those, whether or not they are useful for the downstream task.

The additional de-duplication also had a practical benefit in that the resulting corpus was much smaller, allowing us to perform calculations in memory (e.g., that in §3.2) on the entire corpus at once rather than having to approximate them in mini-batches.

We also discarded sentence pairs that were exactly the same on each side, in which one sentence contained more than 150 tokens, in which the two sentences' numbers did not match, or in which there were suspiciously non-German or non-English sentences according to the pyCLD2 language detector[2]. When pyCLD2 believed a putatively German sentence to be something other than German with certainty greater than 0.5, or a putatively English sentence to be something other than English with certainty greater than 0.5, it was discarded.

### 2.3 Parallelism estimation

With sentence vectors (§2.1) for the reduced corpus (§2.2) in hand, we set out to estimate the degree of parallelism of sentence pairs. A novel measure of parallelism, based on ratios of squared Mahalanobis distances, performed better on a synthetic dataset than some more obvious measurements, and the single-feature submission based on it was our best unsupervised submission.

We also made several other unsupervised measurements:

[2] https://github.com/aboSamoor/pycld2

1. Perplexity of the German sentence according to a 6-gram KenLM language model[3] (Heafield, 2011)
2. Perplexity of the English sentence according to a 6-gram KenLM language model
3. The ratio between (1) and (2), to find sentences pairs that contain different amounts of information
4. Cosine distances between German and English sentence vectors, in a bilingual `sent2vec` space trained only on the target corpus

As we did not have a supervisory signal, we did not have a principled way of choosing weights for these features. Instead, we simply took an unweighted average of the above four features and the Mahalanobis feature in §3.2, after rescaling each to the interval [0.0, 1.0]. As seen in §5, systems based on this feature combination (`NRC-mono-bicov` and `NRC-mono`) were outperformed by our single-feature system in most conditions.

We also considered combinations of these unsupervised measurements with supervised measurements, but this attempt was also unsuccessful compared to a system that used only a single supervised measurement for sentence pair ranking (Lo et al., 2018).

### 2.4 Post-filtering

After scoring each sentence for parallelism, we performed another de-duplication step. In this step, we iterated over each target-language sentence in order of parallelism (that is, sentences assessed to have the highest parallelism were considered first), and removed pairs that only consisted of bigrams that had already been seen. (That is to say, a sentence pair was kept only if it contains a bigram that had not previously been seen.)

This step has to occur *after* quality assessment because, in contrast to regular de-duplication, the sentences in question are not identical; the sentence (and the pair it comes from) may differ in quality from the sentence(s) that make it a duplicate, so we want to keep the *best* such sentence, not just the one that happened to come first in the original corpus.

## 3 Mahalanobis ratios for parallelism assessment

As mentioned in §2.3, we performed several unsupervised measurements on each sentence pair; of these, the measurement that best predicted parallelism (on synthetic data and on our small 300-sentence annotated set) was a novel measurement based on squared Mahalanobis distances.

This measurement rests on two insights:

- If sentence vectors (or in our case, sentence-pair vectors) are normally distributed, the probability that we draw a particular vector (or a more extreme vector) is related to the squared Mahalanobis distance via the $\chi^2$ distribution.
- If the two sentences relate the same information, the probability of drawing the vector for that pair should not be much less than the probability of drawing the individual sentence vectors in isolation.

While Mahalanobis distance is a common statistical measurement, particularly in anomaly detection (e.g. Reed and Yu, 1990), it is not commonly used in machine translation, so we briefly introduce it below.[4]

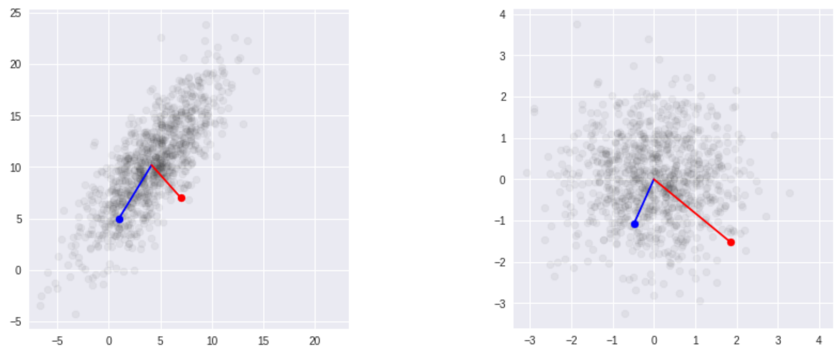
### 3.1 Mahalanobis distance

The probability of a draw from a univariate normal distribution can be related to its distance to the mean in terms of standard deviations (the z-score). In a multivariate normal distribution, however, just measuring the Euclidean distance to the mean can lead to incorrect conclusions; visual inspection of Figure 1a illustrates that the red vector, despite being a clear outlier, is nonetheless closer to the mean than the blue vector.

Rather, the appropriate measurement for relating distance to probability is the square of the Mahalanobis distance (Mahalanobis, 1936); for a vector $x$ from distribution $X$ with correlation $\Sigma$ and mean $\mu$:

[3] Although we assumed that high perplexity sentences would be worse—that they might be ungrammatical, for example—sanity checking suggested higher-perplexity sentences were actually better. Error analysis later suggested that many non-parallel (or parallel but non-informative) sentences were short, possibly explaining why taking perplexity as a *positive* feature resulted in higher scores in sanity-checking.

[4] The following relies heavily on the explanation in Boggs (2014). Note that this explanation is also concerned with the square of the Mahalanobis distance rather than the Mahalanobis distance; it is typical for authors to describe both as "Mahalanobis distance" in prose (cf. Warren et al., 2011, p. 10). It is also typical to use "Mahalanobis distance" to specifically refer to Mahalanobis distance from a point to the mean, although this distance is defined for any two points.



(a) Euclidean distance

(b) Mahalanobis distance

Figure 1: Euclidean distance to the mean in a multivariate normal distribution is not necessarily related to probability; in figure (a), the red vector, despite being an outlier, is closer to the mean. In figure (b), we have rescaled and decorrelated the distribution; Euclidean distance measured in the resulting space (the Mahalanobis distance) can be related to probability through the $\chi^2$ distribution.

$$d^2(x) = (x-\mu)^T\Sigma^{-1}(x-\mu) \tag{1}$$

This is equivalent to decorrelating and rescaling to unit variance in all dimensions, via the inverse square root of the correlation matrix ("Mahalanobis whitening"), and then measuring the squared Euclidean distance to the mean in the resulting space.

$$\begin{aligned} d^2(x) &= (x-\mu)^T\Sigma^{-\frac{1}{2}}\Sigma^{-\frac{1}{2}}(x-\mu) && (2)\\ &= (\Sigma^{-\frac{1}{2}}(x-\mu))^T(\Sigma^{-\frac{1}{2}}(x-\mu)) && (3)\\ &= \|\Sigma^{-\frac{1}{2}}(x-\mu)\|_2^2 && (4) \end{aligned}$$

Figure 1b illustrates the same distribution transformed by $\Sigma^{-\frac{1}{2}}$; we can see that now the magnitude of the outlier red vector is greater than the magnitude of the blue vector.

As mentioned above, the squared magnitudes can be used to calculate probabilities, but in practice the probabilities were so similar in higher-dimensional spaces as to be identical. There remains the possibility, however, that the magnitudes themselves remain sufficiently informative; this was borne out in practice.

### 3.2 Calculating the magnitude ratios

We have high-dimensional vectors, trained monolingually, of German and English sentences (§2.1). We consider their joint distribution by simply concatenating their vectors; there is no additional utility here in learning a translation between the monolingual spaces. We recenter the distribution to have zero mean—this simply makes the calculation and presentation easier—and transform the resulting matrix by $\Sigma^{-\frac{1}{2}}$.

For each sentence vector pair $\langle l_1, l_2\rangle$ (after recentering), we consider three vectors in the transformed space:

- the vector $e_1$ corresponding only to $l_1$'s contribution to the concatenated and transformed vector (as if $l_2 = \vec{0}$)
- the vector $e_2$ corresponding only to $l_2$'s contribution (as if $l_1 = \vec{0}$)
- the vector $e$ corresponding to the transformation of the concatenation of $l_1$ and $l_2$

$$\begin{aligned} e_1 &= \Sigma^{-\frac{1}{2}}(l_1, \vec{0}) && (5)\\ e_2 &= \Sigma^{-\frac{1}{2}}(\vec{0}, l_2) && (6)\\ e &= \Sigma^{-\frac{1}{2}}(l_1, l_2) = e_1 + e_2 && (7) \end{aligned}$$

The measurement $m$ we are interested in is the squared magnitude of the combined vector, divided by the sum of the squared magnitudes of $e_1$ and $e_2$ alone.

$$m = \frac{\|e\|_2^2}{\|e_1\|_2^2 + \|e_2\|_2^2} \tag{8}$$

Roughly speaking, does the sentence pair vector $e$ in Mahalanobis space give more information (expressed in terms of its squared magnitude) than the component sentence vectors $e_1$ and $e_2$ do on their own? If so, we consider them unlikely to

| $p$ | **0.1** | **0.2** | **0.3** | **0.4** | **0.5** |
|---|---|---|---|---|---|
| Mahalanobis | **0.977** | **0.976** | **0.974** | **0.972** | **0.972** |
| Linear | 0.944 | 0.930 | 0.920 | 0.914 | 0.913 |
| Nonlinear | 0.871 | 0.871 | 0.897 | 0.900 | 0.905 |

Table 1: Accuracy of distinguishing parallel (i.e., related by a translation matrix $T$) vs. non-parallel (i.e., random) vectors, from a synthetic dataset of 100,000 pairs of 50-dimensional vectors, plus standard normal additive noise. $p$ represents the proportion of parallel pairs in the dataset.

| $\sigma$ | **1.0** | **2.0** | **3.0** | **4.0** | **5.0** |
|---|---|---|---|---|---|
| Mahalanobis | **.974** | **.778** | **.665** | **.617** | **.597** |
| Linear | .920 | .722 | .640 | .606 | .592 |
| Nonlinear | .897 | .658 | .600 | .586 | .582 |

Table 2: Accuracy of distinguishing parallel (i.e., related by a translation matrix $T$) vs. non-parallel (i.e., random) vectors, from a synthetic dataset of 100,000 pairs of 50-dimensional vectors and "true" proportion $p = 0.3$, with varying degrees of additive noise. $\sigma$ represents the standard deviation of the additive noise added to each of L1 and L2.

be parallel. We take the resulting value $m$ to be the ranking (with lower values being better) for the post-filtering step described in §2.4.

Implementation-wise, we do not actually have to concatenate $l_2$ or $l_1$ with zeros in order to calculate (5) and (6), we can just multiply $l_1$ and $l_2$ by the relevant sub-matrix of $\Sigma^{-\frac{1}{2}}$. It is also unnecessary to actually transform the vector corresponding to the concatenation of $\langle l_1, l_2 \rangle$; the result is just the element-wise sum of $e_1$ and $e_2$.

```
def mahalanobis_whitening(X):
  # inverse square root of covariance
  cov = np.cov(X, rowvar=False)
  inv_cov = np.linalg.inv(cov)
  L, V = np.linalg.eig(inv_cov)
  diag = np.diag(np.sqrt(L))
  return V.dot(diag).dot(V.T)

def ssq(X): # sum of squares
  return np.sum(X*X, axis=1)

def mahalanobis_ratio(L1, L2):
  L1 -= L1.mean(axis=0)
  L2 -= L2.mean(axis=0)
  L = np.concatenate([L1,L2], axis=1)
  whitener = mahalanobis_whitening(L)
  E1 = L1.dot(whitener[:L1.shape[1],:])
  E2 = L2.dot(whitener[L1.shape[1]:,:])
  return ssq(E1+E2) / (ssq(E1) + ssq(E2))
```

Figure 2: Sample implementation of the Mahalanobis ratio calculation in Python, for two $n \times d$ NumPy arrays representing $n$ samples of $d$-dimensional sentence vectors for two languages.

In code, this is a very simple calculation (only about 15 lines of Python+NumPy) and efficient (taking only a few minutes for millions of sentences), provided one has enough system memory to calculate it in one fell swoop. A sample implementation is given in Figure 2.

## 4 Internal results

### 4.1 Synthetic data

The unsupervised measurements on the sentence vectors were first tested on purely synthetic data: two sets of random normal vectors L1 and L2, in which some proportion $p$ of vectors in L1 corresponded to L2 via a linear transformation T, and some proportion of vectors did not. We also added some Gaussian noise to each of L1 and L2, so that this transformation would not be perfect (as it would not be in real data). We varied the proportion of "true" pairs, and the proportion of additive noise, to test how robust these measurements would be in a variety of noise conditions.

Accuracy measurements on this data were made by thresholding scores so that the top $p$ scores are set to 1.0 and the rest to 0.0.[5] This is also how we evaluate accuracy during sanity checking, below.

Table 1 contrasts three systems:

[5] Since the overall task is a *ranking* task, rather than a classification task, we do not at any point have to set a particular threshold for keeping data; this is a way in which the task at hand is easier than a typical anomaly detection task. We therefore simply use the correct proportion to set the thresholds.

1. (**Mahalanobis**) We perform the Mahalanobis ratio calculation described in §3.2.
2. (**Linear**) We learn a linear regression between L1 and L2, transform L1 according the resulting matrix, and measure the cosine similarity between the result and L2.
3. (**Nonlinear**) System (2), but instead of a linear regression we construct a simple two-layer perceptron with a ReLU nonlinearity.[6]

In each condition, the Mahalanobis measurement outperformed the other measurements. It may, of course, be that the conditions of this synthetic data are unlike real data—the relationship between the German and English sentence vectors might, for example, be better approximated with a nonlinear relationship—but, given the comparatively robust performance of the Mahalanobis measurement against a variety of noise conditions, we prioritized our development time to exploring it further.

### 4.2 Sanity checking

We also annotated about 300 random sentence pairs from the target corpus, according to whether we judged them to be parallel or not. We did not tune any parameters to this set, except to make sure that one hyperparameter, the dimensionality of the sentence vectors, did not lead to a numerical underflow condition as dimensionality increased.

Many of our initial attempts at measuring probabilities (and log probabilities) of sentence draws in higher dimensions (e.g. higher than 50) led to the differences between probabilities being so small that they could not be distinguished by floating-point representations, leading to a situation in which almost all probabilities were equivalent and no meaningful comparisons could be made, and thus to random performance when ranking sentences pairs. Keeping the measurements in terms of distances, and not converting them to probabilities, did appear to allow fine-grained comparison in higher dimensions, but we wanted to ensure that continuing to increase the dimensionality did not lead to indistinguishable measurements again.

Sanity checking (Table 3) confirmed that higher dimensionality does not necessarily lead to poorer discrimination: while 10-dimensional vectors only led to 44.1% accuracy in discriminating parallel from non-parallel pairs, 300-dimensional vectors gave 63.4% accuracy.

| Dimensionality | 10 | 50 | 100 | 300 |
|---|---|---|---|---|
| Accuracy | .441 | .548 | .483 | **.634** |

Table 3: Sanity-checking results on 300 annotated sentences, for the Mahalanobis calculation (§3.2) on 10-, 50-, 100-, and 300-dimensional sentence vectors.

It is unclear why 100-dimensional vectors perform more poorly than both 50- and 300-dimensional vectors, but in any case this dataset only has 300 samples and we do not want to put too much stock in the results. The real purpose of this trial was to determine if the curse of dimensionality affects the Mahalanobis measurement adversely, and it does not appear to do so. We therefore used 300-dimensional vectors in our final submissions.

## 5 Official Results

Table 4 presents the results of the official evaluation, on seven corpora in four conditions. To help navigate the wall of numbers, keep in mind that we are mostly interested in the top unsupervised system `NRC-seve-bicov`, and that each table also presents average scores across the seven corpora, in the bottom right corner of each.

In the 100M-word conditions (that is to say, in the conditions where a statistical or neural machine translation system was trained on the top 100M words, as ranked by our filters), we find generally strong performance, with `NRC-seve-bicov` always performing above the median system and with most results in the top 10 (among 48 submissions).

However, we generally observe weaker downstream MT performance in 10M conditions, compared to other competitors; performing roughly near the median system in the NMT 10M condition and frequently below the median in the SMT 10M condition. This suggests to us that the unsupervised systems are adequate in finding

[6] We did not expect this to outperform the linear version—after all, there is no actual nonlinearity in the relationship between L1 and L2—but nonetheless wanted to see how a nonlinear regression would perform in different noise conditions. We observe, for example, that it does unsurprisingly poorly when only a low proportion $p$ of sentences are related, a condition in which a linear regression performs comparatively well.

**SMT, 10M-word**

| | dev. | test | | | | | | |
|---|---|---|---|---|---|---|---|---|
| domain | news | news | speech | laws | medical | news | IT | |
| corpus | newstest17 | newstest18 | iwslt17 | Acquis | EMEA | GlobalVoices | KDE | average |
| top score | **23.23 (1)** | **29.59 (1)** | **22.16 (1)** | **21.45 (1)** | **28.70 (1)** | **22.67 (1)** | **25.51 (1)** | **24.58 (1)** |
| seve-bicov | 19.66 (33) | 25.96 (32) | 18.64 (35) | 18.78 (23) | **27.94 (5)** | 20.05 (28) | 21.38 (41) | 22.13 (29) |
| mono-bicov | 19.61 (35) | 25.13 (36) | 17.86 (39) | 16.59 (35) | 24.21 (37) | 19.97 (34) | 22.07 (37) | 20.97 (38) |
| mono | 17.98 (41) | 23.49 (41) | 16.63 (41) | 15.49 (40) | 23.09 (40) | 18.65 (40) | 21.39 (40) | 19.79 (41) |
| **SMT, 100M-word** | | | | | | | | |
| top score | **25.80 (1)** | **31.35 (1)** | **23.17 (1)** | **22.51 (1)** | **31.45 (1)** | **24.00 (1)** | **26.93 (1)** | **26.49 (1)** |
| seve-bicov | 25.61 (11) | **31.11 (8)** | **22.84 (10)** | 22.19 (15) | **31.20 (3)** | **23.67 (10)** | 26.47 (18) | **26.25 (9)** |
| mono-bicov | **25.65 (5)** | **31.12 (5)** | **22.84 (10)** | **22.37 (8)** | **31.11 (7)** | **23.75 (7)** | 26.19 (30) | **26.23 (10)** |
| mono | 25.45 (14) | 30.63 (21) | 22.72 (20) | 22.06 (21) | 30.74 (20) | **23.70 (9)** | 26.20 (28) | 26.01 (19) |

**NMT, 10M-word**

| | dev. | test | | | | | | |
|---|---|---|---|---|---|---|---|---|
| domain | news | news | speech | laws | medical | news | IT | |
| corpus | newstest17 | newstest18 | iwslt17 | Acquis | EMEA | GlobalVoices | KDE | average |
| top score | **29.44 (1)** | **36.04 (1)** | **25.64 (1)** | **25.57 (1)** | **32.72 (1)** | **26.72 (1)** | **28.25 (1)** | **28.62 (1)** |
| seve-bicov | 24.49 (27) | 30.32 (27) | 21.47 (24) | 22.57 (15) | **31.71 (2)** | 23.08 (27) | 22.89 (27) | 25.34 (21) |
| mono-bicov | 23.38 (30) | 28.86 (32) | 19.33 (34) | 19.03 (29) | 26.45 (32) | 22.03 (32) | 23.72 (23) | 23.07 (30) |
| mono | 20.83 (35) | 24.97 (37) | 17.19 (37) | 16.57 (38) | 23.79 (38) | 19.75 (35) | 21.85 (31) | 20.69 (35) |
| **NMT, 100M-word** | | | | | | | | |
| top score | **32.41 (1)** | **39.85 (1)** | **27.43 (1)** | **28.43 (1)** | **36.72 (1)** | **29.26 (1)** | **30.92 (1)** | **32.06 (1)** |
| seve-bicov | **32.10 (2)** | **39.39 (7)** | **27.09 (6)** | **28.31 (5)** | **36.30 (10)** | **28.94 (9)** | 30.12 (16) | **31.69 (8)** |
| mono-bicov | **31.67 (9)** | 38.86 (15) | **27.10 (5)** | **28.15 (9)** | 35.96 (15) | 28.87 (11) | 30.41 (11) | 31.56 (11) |
| mono | 31.39 (16) | 38.42 (21) | 26.80 (12) | 27.94 (12) | 35.71 (21) | 28.00 (27) | 30.32 (14) | 31.20 (19) |

Table 4: BLEU scores (and ranking, out of 48 submissions) of NRC's unsupervised submissions: "seve" indicates single-feature (Mahalanobis ratio) parallelism assessment, "mono" indicates parallelism assessment using an unweighted ensemble of unsupervised features, "bicov" indicates that the final bigram coverage step (§2.4) was performed. Results in the top 10 performers are bolded.

a 100M word training set[7] but relatively poor at sub-selecting higher-quality sentences from that set. We think this may be because our system might have a bias towards picking relatively similar sentences, rather than the more diverse set of sentences that an MT training set needs, which is amplified in the 10M condition.

A surprising exception to this weakness is the European Medicines Agency (EMEA) corpus, in which `NRC-seve-bicov` is the #5 and #2 system in the SMT 10M and NMT 10M conditions, respectively. This could suggest that competitors are overfitting to the domain(s) of the training data, and performing correspondingly poorly on the out-of-domain EMEA, whereas `NRC-seve-bicov` cannot overfit in this manner. However, the other NRC unsupervised submissions, which also cannot overfit, have no special advantage on EMEA, and nor does `NRC-seve-bicov` perform notably well on other out-of-domain corpora in the 10M conditions.

## 6 Future research

The unsupervised methods described here seem promising in distinguishing parallel from non-parallel sentence pairs, but we interpret the 10M-word results as suggesting they are comparatively poor at distinguishing other MT-relevant features of sentence-pair quality. Considering bigram coverage (§2.4) appears to help somewhat, but more research is needed into mitigating the tendency of these measurements to prefer an uninteresting selection of sentences.

Also, it is likely that a sentence-vector, even a high-dimensional one, is not sufficiently fine-grained to choose the highest-quality pairs; the process described in this paper essentially says that two sentences with sufficiently similar topics are to be considered parallel, even if there is little word-level correlation between the sentences. We therefore intend to investigate a word-level analogue of the sentence-level Mahalanobis ratio measurement.

[7] Spot-checking a random sample of sentences suggested to us that there were indeed roughly 100M words worth of genuinely parallel data, but much of it would not have been particularly informative for machine translation. We therefore interpret 100M results as representing one's success at identifying parallel data, and the 10M results as representing how well one assesses usefulness-for-MT beyond parallelism.

## References

Thomas Boggs. 2014. Whitening characteristics of the Mahalanobis distance. http://blog.bogatron.net/blog/2014/03/11/mahalanobis-whitening/.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *MT Summit 2005*.

Philipp Koehn, Kenneth Heafield, Mikel L. Forcada, Miquel Esplà-Gomis, Sergio Ortiz-Rojas, Gema Ramírez Sánchez, Víctor M. Sánchez Cartagena, Barry Haddow, Marta Bañón, Marek Střelec, Anna Samiotou, and Amir Kamran. 2018a. ParaCrawl corpus version 1.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018b. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Chi-kiu Lo, Michel Simard, Darlene Stewart, Samuel Larkin, Cyril Goutte, and Patrick Littell. 2018. Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The NRC supervised submissions to the parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation (WMT 2018)*.

Prasanta Chandra Mahalanobis. 1936. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2:49–55.

Joel Martin, Howard Johnson, Benoît Farley, and Anna Maclachlan. 2003. Aligning and using an English-Inuktitut parallel corpus. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: Data driven machine translation and beyond, Volume 3*, pages 115–118. Association for Computational Linguistics.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540. Association for Computational Linguistics.

Irving S Reed and Xiaoli Yu. 1990. Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(10):1760–1770.

Jörg Tiedemann. 2009. News from OPUS: A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Rik Warren, Robert F Smith, and Anne K Cybenko. 2011. Use of Mahalanobis distance for detecting outliers and outlier clusters in markedly non-normal data: A vehicular traffic example. Technical report, SRA International Inc., Dayton, OH.

# Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The NRC supervised submissions to the Parallel Corpus Filtering task

Chi-kiu Lo Michel Simard Darlene Stewart
Samuel Larkin Cyril Goutte Patrick Littell
NRC-CNRC
Multilingual Text Processing
National Research Council Canada
1200 Montreal Road, Ottawa, ON K1A 0R6, Canada
Firstname.Lastname@nrc-cnrc.gc.ca

## Abstract

We present our semantic textual similarity approach in filtering a noisy web crawled parallel corpus using YiSi—a novel semantic machine translation evaluation metric. The systems mainly based on this supervised approach perform well in the WMT18 Parallel Corpus Filtering shared task (4th place in 100-million-word evaluation, 8th place in 10-million-word evaluation, and 6th place overall, out of 48 submissions). In fact, our best performing system—NRC-yisi-bicov is one of the only four submissions ranked top 10 in both evaluations. Our submitted systems also include some initial filtering steps for scaling down the size of the test corpus and a final redundancy removal step for better semantic and token coverage of the filtered corpus. In this paper, we also describe our unsuccessful attempt in automatically synthesizing a noisy parallel development corpus for tuning the weights to combine different parallelism and fluency features.

## 1 Introduction

The WMT18 shared task on parallel corpus filtering (Koehn et al., 2018b) challenged teams to find clean sentence pairs from ParaCrawl, a humongous high-recall, low-precision web crawled parallel corpus (Koehn et al., 2018a), for training machine translation (MT) systems. Data cleanliness of parallel corpora for MT systems is affected by a wide range of factors, e.g., the parallelism of the sentence pairs, the fluency of the sentences in the output language, etc. Previous work (Goutte et al., 2012; Simard, 2014) showed that different types of errors in the parallel training data degrade MT quality at different levels. Intuitively, the crosslingual semantic textual similarity of the sentence pairs in the corpora is one of the most important factors affecting the parallelism of the target sentence pairs. Lo et al. (2016) scored crosslingual semantic textual similarity crosslingually, using a semantic MT quality estimation metric with fewer resource requirements, or monolingually, using a pipeline of MT system and semantic MT evaluation metric with better performance. The core of the National Research Council of Canada (NRC) supervised submissions (NRC-yisi-bicov and NRC-yisi) of the parallel corpus filtering shared task were developed in the same philosophy using a new semantic MT evaluation metric, YiSi (Lo, 2018).

The participants of the parallel corpus filtering shared task were given a large set of "clean" German-English monolingual and bilingual training corpora for the WMT18 news translation shared task (except a filtered version of ParaCrawl) and tasked to score the cleanliness of each sentence pair in the "dirty" ParaCrawl corpus. Our supervised submissions used the given parallel data to train an MT system to translate the German side of the dirty corpus into English. The provided version of the dirty ParaCrawl corpus contains raw data crawled from the web with minimal de-duplication processing only, and includes non-parallel, or even non-linguistic data. It contains 104 million German-English sentence pairs, with 1 billion English tokens and 964 million German tokens before punctuation tokenization. A 10-million-word (10M-word) and a 100-million-word (100M-word) corpus sub-selected by the participating cleanliness scoring system were used to train statistical machine translation (SMT) and neural machine translation (NMT) systems. The success of the participating scoring systems was determined by the quality of the MT output from the four MT systems as measured by BLEU (Papineni et al., 2002) on some in-domain and out-of-domain evaluation sets.

In this paper, we describe the efforts in developing our supervised submissions: the initial fil-

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 908–916
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64108

tering steps for scaling down the size of the given ParaCrawl dirty corpus, the wide range of features experimented for measuring parallelism, fluency and grammaticality, the failed attempt to combine useful features and the final redundancy removal for improving token coverage of the filtered corpus. Despite the simple single-feature architecture used in the NRC best-performing supervised submission (`NRC-yisi-bicov`), it performed well in the MT quality evaluation compared to other participants. It ranked 4th in the 100-million-word evaluation, 8th in the 10-million-word evaluation and 6th overall among 48 submissions. It is one of the only four submissions ranked top 10 in both evaluations.

## 2 System architecture

There are a wide range of factors constituting a good parallel sentence pair for training MT systems. Some of the more important factors for a good general MT system parallel training corpus include:

- High parallelism in the sentence pairs
- High fluency and grammaticality, especially for sentences in the output language
- High token coverage, especially in the input language
- High variety of sentence lengths

The NRC supervised and unsupervised submissions shared the same general skeleton for the system architecture. The systems consisted of: initial filtering to remove obvious noise and to prevent selections constituted of a large collection of short sentences; feature scoring for measuring parallelism, fluency and grammaticality; feature combination (only in the `NRC-mono` and `NRC-mono-bicov` submissions); and final redundancy removal (only in the `NRC-⋆-bicov` submissions) to improve token coverage.

### 2.1 Initial filtering

Although the given "dirty" corpus had already been deduplicated, we did an additional deduplication step in which email and web addresses were replaced with a placeholder token, before deciding which sentences were duplicates. Sentence pairs were filtered out if the pair was seen before or if the input side was exactly the same as the output side.

We also observed that many sentences in the corpus, although parallel, were rather similar and uninformative, especially numerical data such as long lists of page numbers or dates. We observed that using measurements that preferred such sentences resulted in comparatively poor MT performance, likely because the MT systems did not get enough varied data. To mitigate this, we ran two additional filtering steps regarding numbers. First, over 50% of the numbers on each side of the sentence pair had to have a match, otherwise it was filtered out as a bad translation. Next, we removed all the numbers and punctuation and, similar to the previous deduplication step, filtered out sentence pairs if their non-number parts had been seen before, or if the non-number input side was exactly the same as the non-number output side.

A common error found in web crawled corpora is sentences that are in the wrong language. We therefore ran the `pyCLD2` language detector[1] on each side of the sentence pair and filtered out pairs whose input side was non-German with a confidence score over 0.5, or whose output side was non-English with a confidence score over 0.5.

Our final filtering step was to remove unreasonably long sentences. Another common error in web crawled corpora is that they contain nonlinguistic data, such as tables or computer code. We therefore punctuation-tokenized both sides of the sentence pairs and removed the pair if either side was more than 150 tokens.

The above mentioned steps removed obvious and uninteresting noise and significantly scaled down the size of the original ParaCrawl corpus for more resource demanding feature scoring. The corpus was scaled down from 104 million sentence pairs originally to 28 million sentence pairs.

### 2.2 Feature scoring

We experimented with a large collection of feature models to address the factors for good general MT training data mentioned at the beginning of this section. Below is a selected list of features that performed reasonably well in our internal sanity check.

#### 2.2.1 Parallelism

**YiSi-1: monolingual semantic MT evaluation metric** We first used the "clean" WMT18 news translation task monolingual and parallel training data (tokenized and lowercased) to train an

[1] `https://github.com/aboSamoor/pycld2`

SMT system using Portage (Larkin et al., 2010), a conventional log-linear phrase-based SMT system. The translation model of the SMT system uses IBM4 word alignments (Brown et al., 1993) with grow-diag-final-and phrase extraction heuristics (Koehn et al., 2003). The system has two n-gram language models: a 5-gram mixture language model (LM) trained on the four corpora components using SRILM (Stolcke, 2002), and a pruned 6-gram LM trained on the WMT monolingual English training corpus built using KenLM (Heafield, 2011). The SMT system also includes a hierachical distortion model, a sparse feature model consisting of the standard sparse features proposed in Hopkins and May (2011) and sparse hierarchical distortion model features proposed in Cherry (2013), and a neural network joint model, or NNJM, with 3 words of target context and 11 words of source context, effectively a 15-gram LM (Vaswani et al., 2013; Devlin et al., 2014). The parameters of the log-linear model were tuned by optimizing BLEU on the development set (newstest2017) using the batch variant of margin infused relaxed algorithm (MIRA) by Cherry and Foster (2012). Decoding uses the cube-pruning algorithm of Huang and Chiang (2007) with a 7-word distortion limit. We then translated the German side of the filtered ParaCrawl into English.

We also used the monolingual English data to train word embeddings using `word2vec` (Mikolov et al., 2013) for evaluating monolingual lexical semantic similarity.

YiSi is new a semantic MT evaluation metric inspired by MEANT 2.0 (Lo, 2017). YiSi-1 is equivalent to MEANT 2.0-nosrl. It measures the segmental semantic similarity. The segmental semantic precision and recall divide the inverse-document-frequency weighted sum of the n-gram lexical semantic similarity of the MT output and the English sentence of the target pair by the weighted count of n-grams in the MT output and the English sentences, respectively. In this work, we set the n-gram size to two. Precisely, YiSi-1 is computed as follows:

$$\begin{aligned}
w(e) &= \text{inverse document freq. of token } e \\
w(\overrightarrow{e}) &= \sum_k w(e_k) \\
v(e) &= \text{word embedding of token } e \\
s(e, f) &= cos(v(e), v(f))
\end{aligned}$$

$$s_p(\overrightarrow{e}, \overrightarrow{f}) = \frac{\sum_a w\left(\overrightarrow{e_{a..a+n-1}}\right) \cdot \max\limits_b \frac{\sum_{k=0}^{n-1} w(e_{a+k}) \cdot s(e_{a+k}, f_{b+k})}{\sum_{k=0}^{n-1} w(e_{a+k})}}{\sum_a w\left(\overrightarrow{e_{a..a+n-1}}\right)}$$

$$s_r(\overrightarrow{e}, \overrightarrow{f}) = \frac{\sum_b w\left(\overrightarrow{f_{b..b+n-1}}\right) \cdot \max\limits_a \frac{\sum_{k=0}^{n-1} w(f_{b+k}) \cdot s(e_{a+k}, f_{b+k})}{\sum_{k=0}^{n-1} w(f_{b+k})}}{\sum_b w\left(\overrightarrow{f_{b..b+n-1}}\right)}$$

$$\begin{aligned}
\text{precision} &= s_p(\overrightarrow{e_{\text{sent}}}, \overrightarrow{f_{\text{sent}}}) \\
\text{recall} &= s_r(\overrightarrow{e_{\text{sent}}}, \overrightarrow{f_{\text{sent}}}) \\
\text{YiSi-1} &= \frac{\text{precision} \cdot \text{recall}}{\alpha \cdot \text{precision} + (1-\alpha) \cdot \text{recall}}
\end{aligned}$$

YiSi-1_srl measures the semantic similarity with additional frame semantic or semantic role labeling (srl) information. It uses a more principle way to compute the precision and recall of semantic similarity between the translation output and the reference when comparing to MEANT 2.0. Instead of aggregating the precision and recall at the segmental semantic similarity level, YiSi-1_srl precision is the weighted sum of the segmental semantic precision and the frame semantic precision and similarly, for YiSi-1_srl recall. The frame semantic precision is the weighted sum of the segmental semantic precision of the semantic role fillers according to the shallow semantic structure parsed by the `mateplus` (Roth and Woodsend, 2014) English semantic parser over the weighted counts of roles and frames according to the shallow semantic structure of the MT output and similarly, for the frame semantic recall. Precisely, YiSi-1_srl is computed as follows:

$$\begin{aligned}
q^0_{i,j} &= \text{ARG } j \text{ of aligned frame } i \text{ in MT} \\
q^1_{i,j} &= \text{ARG } j \text{ of aligned frame } i \text{ in REF} \\
w^0_i &= \frac{\text{\#tokens filled in aligned frame } i \text{ of MT}}{\text{total \#tokens in MT}} \\
w^1_i &= \frac{\text{\#tokens filled in aligned frame } i \text{ of REF}}{\text{total \#tokens in REF}} \\
w_j &= \text{count}\,(\text{ARG } j \text{ in REF}) \\
w_t &= 0.25 * \text{count}\,(\text{predicate in REF}) \\
\text{srl}_p &= \frac{\sum_i w^0_i \frac{w_t s_p(\overrightarrow{e_{i,t}}, \overrightarrow{f_{i,t}}) + \sum_j w_j s_p(\overrightarrow{e_{i,j}}, \overrightarrow{f_{i,j}})}{w_t + \sum_j w_j |q^0_{i,j}|}}{\sum_i w^0_i} \\
\text{srl}_r &= \frac{\sum_i w^1_i \frac{w_t s_r(\overrightarrow{e_{i,t}}, \overrightarrow{f_{i,t}}) + \sum_j w_j s_r(\overrightarrow{e_{i,j}}, \overrightarrow{f_{i,j}})}{w_t + \sum_j w_j |q^1_{i,j}|}}{\sum_i w^1_i}
\end{aligned}$$

$$\begin{aligned}
\text{precision} &= \beta \cdot \text{srl}_p + (1-\beta) \cdot s_p(\overrightarrow{e_{\text{sent}}}, \overrightarrow{f_{\text{sent}}}) \\
\text{recall} &= \beta \cdot \text{srl}_r + (1-\beta) \cdot s_r(\overrightarrow{e_{\text{sent}}}, \overrightarrow{f_{\text{sent}}})
\end{aligned}$$

$$\text{YiSi-1\_srl} = \frac{\text{precision} \cdot \text{recall}}{\alpha \cdot \text{precision} + (1-\alpha) \cdot \text{recall}}$$

When we evaluate MT output in practice, YiSi score is a weighted harmonic mean of the precision and recall. However, in this work, we segregated the precision and recall of YiSi into separate features as we planned to let the regression decide suitable weights to combine them. Further details of YiSi are provided in Lo (2018).

**YiSi-2: crosslingual semantic MT evaluation metric** For the crosslingual version of YiSi, YiSi-2, instead of training a German-English MT system, we used the "clean" WMT18 news translation task parallel training data to train bilingual word embeddings using `bivec` (Luong et al., 2015) for evaluating crosslingual lexical semantic similarity.

Similar to YiSi-1, YiSi-2 precision and recall are the weighted sum of the crosslingual lexical semantic similarity of the sentence pairs over the weighted count of tokens in the German and English sentences respectively. In this work, we set the n-gram size to one.

YiSi-2_srl precision and recall are the weighted sum of the crosslingual lexical semantic similarity according to the shallow semantic structure parsed by `mateplus` German and English semantic parser over the weighted counts of roles and frames according to the shallow semantic structure of the German and the English sentence, respectively. We also segregated the precision and recall of YiSi-2 and YiSi-2_srl into separate features for the same reason mentioned above.

**Alignment scores** The SMT model trained on the "clean" WMT18 news translation task parallel training data for YiSi score computation include several alignment models as components, from which probabilities $p(d|e)$ and $p(e|d)$ were computed. We find the hidden markov model (HMM) alignment models (Vogel et al., 1996) are reliably useful for scoring parallelism of the sentence pairs in the target corpus.

**Perplexity ratio of input sentences and output sentences** The perplexity ratio reflects the different amounts of information contained in each side of the sentence pairs. This is computed by dividing the smaller perplexity score of the two sentences in the target pair by the larger one. Thus, the ratio ranged from 0 to 1, where a larger value represents better parallelism.

**Perplexity ratio of the part-of-speech (POS) tags of the input sentences and output sentences** Similar to the previous feature, the perplexity ratio of the input and output sentences POS tags is computed by dividing the smaller POS perplexity score of the two sentences in the target pair by the larger one.

**Distance of sentence vectors** Sentence vectors were trained using `sent2vec` (Pagliardini et al., 2018) on each side of the "clean" parallel WMT18 news translation task parallel training data. Further details on how to compute these features are described in Littell et al. (2018).

#### 2.2.2 Fluency and grammaticality

**Perplexity** 6-gram LMs of the input and output languages were built using KenLM (Heafield, 2011) on the WMT18 news translation task German (263 million sentences) and English (303 million sentences) monolingual corpora.

**Perplexity of POS tags** We parsed the German and the English monolingual training data using `mateplus` and built 6-gram LMs based on the POS tags using KenLM.

### 2.3 Feature combination

#### 2.3.1 Synthetic noisy data generation

We used the WMT09-13 test sets (Callison-Burch et al., 2009, 2010, 2011, 2012; Bojar et al., 2013) as the basis of our development set, as we believe that all the test sets in the previous years are clean and highly parallel, as opposed to the "clean" training data where glitches may occur (especially in the Europarl and CommonCrawl corpora). We introduced several types of synthetic errors into the development set as negative examples and assigned scores according to the severity of each error.

We added the output from the best and the worst participating systems in each year as the mostly parallel but less fluent sentence pairs. We also constructed error sentence pairs by offsetting or deleting tokens on either side, or introducing tokens in the wrong language. The target scores of these pairs are proportional to the percentage of tokens offset, deleted or introduced. Lastly, misaligned sentence pairs were added as fluent but non-parallel negative examples. The resulting development set had 11k sentence pairs of positive and synthetic negative examples.

#### 2.3.2 Regression

In order to benefit from multiple features, we first experimented with linear feature combination. Using the scores generated in §2.2 as features, and the data described in the previous section as modeling data, we trained a linear model with $L_1$ regularization. The amount of regularization was set by optimizing a 10-fold cross-validation estimator of the generalization error on the modeling data. On the synthetic data, it turns out that the optimal level of regularization is minimal, suggesting the overfitting is minimal with this amount of data. We also tried building a linear combination of a subset of the most relevant features, selected from the results of the regularized model built on the full set of features (essentially removing features for which combination weights were not significantly different from zero). The linear features combination models yield marginal improvements according to the cross-validation estimator built from the synthetic data. However, there was no gain in precision when evaluated on our small annotated set or in MT quality when training MT system using data sub-selected by the combined model, so we ended up not submitting the combined results.

### 2.4 Redundancy filtering

Our scoring mechanisms naturally tend to assign higher scores to semantically similar sentences without paying attention to their usefulness for MT. As a result, we observe much redundancy and a somewhat limited vocabulary coverage in the top-ranking sentences, such as numerous perfectly translated dateline. To compensate for this effect, we applied a form of redundancy filtering after scoring sentence pairs: going down the re-ranked corpus, we filtered out any sentence pair that did not contain at least one “new” source-language word bigram, i.e., a pair of consecutive source-language tokens not observed in previous pairs. This had the effect of excluding sentences that were too similar to one another. Because it was applied post-scoring on the re-ranked corpus, it tended to retain higher-scoring sentence pairs.

## 3 Experiments and results

### 3.1 Sanity check

We annotated about 300 random sentence pairs from the filtered target corpus, labeling 93 as correct translations and the rest as non-parallel. We did not tune any parameters to this set, since it was small and also doing so would violate the competition guidelines, but used it to sanity check our feature engineering. We computed the precision of each experimented feature by dividing the number of true positives in the top 93 pairs (scored by the feature) by 93.

| **features** | **precision** |
|---|---|
| **baselines** | |
| random | 0.312 |
| hunalign | 0.624 |
| **parallelism** | |
| YiSi-1 precision | **0.796** |
| YiSi-1 recall | 0.763 |
| YiSi-1_srl ($\beta$=1) precision | 0.559 |
| YiSi-1_srl ($\beta$=1) recall | 0.559 |
| YiSi-2 precision | 0.753 |
| YiSi-2 recall | 0.731 |
| YiSi-2_srl ($\beta$=1) precision | 0.441 |
| YiSi-2_srl ($\beta$=1) recall | 0.452 |
| HMM $p(d\|e)$ | 0.753 |
| HMM $p(e\|f)$ | 0.753 |
| s2v d100 cosine | 0.435 |
| s2v d300 Mahalanobis | 0.634 |
| perplexity ratio | 0.538 |
| POS perplexity ratio | 0.441 |
| **fluency and grammaticality** | |
| German perplexity | 0.419 |
| English perplexity | 0.355 |
| German POS perplexity | 0.376 |
| English POS perplexity | 0.462 |
| **feature combination** | |
| regression | 0.763 |

Table 1: Precision on the 300-annotated sentence pairs.

Table 1 shows the precision of the experimented features. We also include the results from a random scoring baseline and the given hunalign scores (Initial filtering was integrated into both baselines). YiSi-1 precision was the best performing feature with close to 80% true positive rate in its top ranking sentence pairs. In general, we can see that supervised parallelism features achieved over 73% precision. It is expected that the structural semantic options of YiSi were less accurate as standalone features due to the fact the score for a sentence pair would be zero when the shallow semantic parser failed to find a semantic frame on either side. Our original plan was to combine these features with other semantic features and bias the combined scores to prefer longer sentences with

| | SMT | | | | NMT | | | |
|---|---|---|---|---|---|---|---|---|
| | 10M-word | | 100M-word | | 10M-word | | 100M-word | |
| system | dev. | test | dev. | test | dev. | test | dev. | test |
| random | 17.52 | 20.28 | 22.06 | 26.88 | 19.58 | 24.06 | 27.27 | 34.63 |
| HMM $p(e\|f)$ | 19.09 | 23.55 | 24.42 | 29.73 | 21.16 | 26.59 | 31.53 | 39.52 |
| HMM $p(e\|f)$ bicov | 20.42 | 25.31 | 24.68 | 29.98 | 23.17 | 29.08 | 31.98 | 39.66 |
| YiSi-1 precision (`NRC-yisi`) | 21.56 | 24.68 | 24.47 | 30.10 | 24.24 | 30.75 | 32.49 | 40.27 |
| YiSi-1 precision bicov (`NRC-yisi-bicov`) | **22.19** | **27.41** | **24.84** | **30.46** | **26.69** | **33.56** | **33.20** | **40.98** |
| regression bicov | 21.86 | 26.97 | **24.84** | 30.27 | 25.28 | 31.94 | 31.30 | 39.34 |

Table 2: BLEU scores of SMT and NMT systems trained on the 10M- and 100M-word corpora subselected by the scoring systems. "bicov" indicates that the final bigram coverage step (§2.4) was performed. The development set is newstest2017 and the test set is newstest2018.

semantic structure recognized by the parser. However, as we can see, the regression hurt the precision on the 300-annotated subset of data. This was the first hint that our feature combination was not a promising avenue.

### 3.2 MT quality check

We used the official software to extract the 10M-word and 100M-word corpora from the original ParaCrawl according to the feature scores. We then trained SMT and NMT systems using the extracted data. The SMT systems were trained using Portage with components and parameters similar to the German-English SMT system in Williams et al. (2016). The NMT systems were transformer models with self-attention (Vaswani et al., 2017) trained using Sockeye-1.18.20 (Hieber et al., 2017) with default parameter settings[2], except for the maximum sequence length, which was reduced to 60:60, and we also clip gradients to 1. We used newstest2017 and newstest2018 as the MT development and test set.

Table 2 shows the BLEU scores for MT systems trained on the ParaCrawl data subselected by our scoring features. We have also included the random scoring feature (with initial filtering) as a baseline. The MT quality trained on data subselected by the feature scores showed the same trend as the results of the sanity check. That is to say, a feature that performed better in the sanity check indeed was able to pick "cleaner" data to train better MT systems.

We noticed that the differences in BLEU of MT systems trained on the 100M-word corpus subselected by our features were very small. This shows that our supervised features were successful in identifying parallel data.

In addition, the results on MT quality confirmed again that our feature combination was not performing as planned. Compared to the systems trained on data subselected by the best feature (YiSi-1 precision bicov), those trained on data subselected by the regression score list had their performance decreased by 0.2-0.5 BLEU on SMT and 1.6 BLEU on NMT.

Systems in which we applied redundancy removal are labeled "bicov". On the larger (100M words) selections, the redundancy removal had virtually no effect when applied after YiSi scoring. However, on the smaller (10M words) selection, it allowed for substantial BLEU score increases: +1.61 BLEU for SMT systems on average and +2.44 BLEU for NMT systems.

## 4 Official Results

Table 3 presents the results of the official BLEU scores on seven development and test sets (devtests) in four training conditions, the average scores across the seven devtests for each of the four training conditions, the average scores across all the devtests for the 10M-word and 100M-word training conditions and the average scores across all the test documents and all training conditions. Our best performing supervised submission—`NRC-yisi-bicov` ranked 4th in the 100M-word evaluation, 8th in the 10M-word evaluation and 6th overall, out of 48 submissions. In fact, it is one of the only four submissions ranked top 10 in all four training conditions.

Our supervised systems perform strongly on the 100M-word conditions with most of the results in the top 10 (among 48 submissions) and very small differences from the highest score of each test set. Similar to the results from our internal MT quality check, the performance differences of our supervised systems on the 100M-word conditions were very small. In other words, the redundancy re-

[2] `https://github.com/awslabs/sockeye/blob/arxiv_1217/arxiv/code/transformer/sockeye/train-transformer.sh`

**SMT**

| | dev. | test | | | | | | |
|---|---|---|---|---|---|---|---|---|
| domain | news | news | speech | laws | medical | news | IT | |
| system \ test set | newstest17 | newstest18 | iwslt17 | Acquis | EMEA | Global Voices | KDE | average |
| | | | | **10M-word** | | | | |
| highest scores | **23.23 (1)** | **29.59 (1)** | **22.16 (1)** | **21.45 (1)** | **28.70 (1)** | **22.67 (1)** | **25.51 (1)** | **24.58 (1)** |
| NRC-yisi-bicov | **22.03 (8)** | **28.72 (6)** | **21.34 (7)** | 19.66 (12) | 26.35 (21) | **22.06 (4)** | **25.21 (3)** | **23.89 (6)** |
| NRC-yisi | 21.34 (20) | 27.97 (12) | **21.12 (9)** | 19.26 (19) | 26.00 (22) | **21.79 (8)** | **24.99 (5)** | **23.52 (10)** |
| | | | | **100M-word** | | | | |
| highest scores | **25.80 (1)** | **31.35 (1)** | **23.17 (1)** | **22.51 (1)** | **31.45 (1)** | **24.00 (1)** | **26.93 (1)** | **26.49 (1)** |
| NRC-yisi-bicov | **25.76 (3)** | **31.35 (1)** | 22.80 (15) | **22.36 (9)** | **31.11 (7)** | **23.84 (5)** | **26.93 (1)** | **26.40 (5)** |
| NRC-yisi | **25.63 (7)** | **31.04 (9)** | **23.16 (2)** | **22.46 (5)** | 30.83 (18) | **23.93 (3)** | **26.82 (5)** | **26.37 (6)** |

**NMT**

| | dev. | test | | | | | | |
|---|---|---|---|---|---|---|---|---|
| domain | news | news | speech | laws | medical | news | IT | |
| system \ test set | newstest17 | newstest18 | iwslt17 | Acquis | EMEA | Global Voices | KDE | average |
| | | | | **10M-word** | | | | |
| highest scores | **29.44 (1)** | **36.04 (1)** | **25.64 (1)** | **25.57 (1)** | **32.72 (1)** | **26.72 (1)** | **28.25 (1)** | **28.62 (1)** |
| NRC-yisi-bicov | **27.61 (8)** | **33.93 (9)** | **24.37 (9)** | 23.20 (12) | 29.75 (13) | **25.44 (7)** | **27.75 (4)** | **27.41 (8)** |
| NRC-yisi | 26.62 (11) | 32.72 (12) | 23.89 (11) | 22.22 (19) | 28.55 (19) | 24.83 (12) | **26.81 (8)** | 26.50 (12) |
| | | | | **100M-word** | | | | |
| highest scores | **32.41 (1)** | **39.85 (1)** | **27.43 (1)** | **28.43 (1)** | **36.72 (1)** | **29.26 (1)** | **30.92 (1)** | **32.06 (1)** |
| NRC-yisi-bicov | **31.97 (3)** | **39.59 (4)** | **26.95 (9)** | **28.35 (4)** | **36.59 (3)** | **29.09 (3)** | **30.70 (5)** | **31.88 (4)** |
| NRC-yisi | 31.53 (11) | **39.30 (9)** | **27.13 (4)** | 27.91 (13) | 36.28 (12) | **29.01 (6)** | **30.92 (1)** | **31.76 (6)** |

| system | 10M-word average | 100M-word average | all average |
|---|---|---|---|
| highest scores | **26.54 (1)** | **29.27 (1)** | **27.90 (1)** |
| NRC-yisi-bicov | **25.65 (8)** | **29.14 (4)** | **27.39 (6)** |
| NRC-yisi | 25.01 (11) | **29.07 (5)** | **27.04 (9)** |

Table 3: BLEU scores (and ranking, out of 48 submissions) of NRC's supervised submissions: "bicov" indicates that the final bigram coverage step (§2.4) was performed. The highest scores of each testing conditions are included for reference. Results in the top 10 performers are bolded.

moval had virtually no effect on the larger selections.

Compared to other top-ranking submissions, both of our supervised submissions have weaker MT performance in the 10M-word training conditions although still rank above the median system on all test sets. This suggests that our systems are generally good at identifying parallel sentences for the 100M-word training set but relatively weaker at ranking the sentence pairs according to the usefulness-for-MT beyond parallelism. Although the redundancy removal heuristic appeared to play a more significant role in the 10M-word training conditions, the improvements on the official test sets are less substantial than what we observed in our internal experiments. This is potentially due to the differences in architecture between our MT systems and the MT systems built in the official evaluation.

## 5 Conclusion

In this paper, we presented the NRC supervised submissions (`NRC-yisi-bicov` and `NRC-yisi`) to the WMT18 parallel corpus filtering task. The core of the submissions used YiSi – a novel semantic machine translation (MT) evaluation metric to score the semantic textual similarity between the translated German side and the English of the target sentence pair. Despite failing to combine with other fluency or grammaticality oriented features, the YiSi-based system with redundancy removal performed well in the shared task, particularly in the 100M-word evaluation (4th place out of 48 submitted systems). This shows that using an adequacy oriented scoring measure is a reliable method to identify good sentence pairs for training MT systems. At the same time, the slightly worse performance in the 10M-word evaluation (8th place out of 48 submitted systems) also suggests that fluency or grammaticality oriented features are useful for fine-grained ranking of MT training data quality. Thus, future work includes investigating other feature combination methodologies, such as more realistic tuning example generation.

## References

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden. Association for Computational Linguistics. Revised August 2010.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland. Association for Computational Linguistics.

Colin Cherry. 2013. Improved reordering for phrase-based translation using sparse features. In *Proceedings of NAACL HLT 2013*.

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proc. 2012 Conf. of the N. American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1370–1380, Baltimore, Maryland.

Cyril Goutte, Marine Carpuat, and George Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas*.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1712.05690*.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362. Association for Computational Linguistics.

Liang Huang and David Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proc. 45th Annual Meeting of the Assoc. for Comp. Linguistics*, pages 144–151, Prague, Czech Republic.

Philipp Koehn, Kenneth Heafield, Mikel L. Forcada, Miquel Esplà-Gomis, Sergio Ortiz-Rojas, Gema Ramírez Sánchez, Víctor M. Sánchez Cartagena, Barry Haddow, Marta Bañón, Marek Střelec, Anna Samiotou, and Amir Kamran. 2018a. ParaCrawl corpus version 1.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018b. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. 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*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Samuel Larkin, Boxing Chen, George Foster, Uli Germann, Eric Joanis, J. Howard Johnson, and Roland Kuhn. 2010. Lessons from NRC's Portage System at WMT 2010. In *5th Workshop on Statistical Machine Translation (WMT 2010)*, pages 127–132.

Patrick Littell, Samuel Larkin, Darlene Stewart, Michel Simard, Cyril Goutte, and Chi-kiu Lo. 2018. Measuring sentence parallelism using Mahalanobis distances: The NRC unsupervised submissions to the WMT18 Parallel Corpus Filtering shared task. In

*Proceedings of the Third Conference on Machine Translation (WMT 2018)*.

Chi-kiu Lo. 2017. MEANT 2.0: Accurate semantic MT evaluation for any output language. In *Proceedings of the Second Conference on Machine Translation*, pages 589–597, Copenhagen, Denmark. Association for Computational Linguistics.

Chi-kiu Lo. 2018. The NRC metric submission to the WMT18 metric and parallel corpus filtering shared task. In *Arxiv*.

Chi-kiu Lo, Cyril Goutte, and Michel Simard. 2016. CNRC at Semeval-2016 task 1: Experiments in crosslingual semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 668–673.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *NAACL Workshop on Vector Space Modeling for NLP*, Denver, United States.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pages 311–318, Philadelphia, Pennsylvania.

Michael Roth and Kristian Woodsend. 2014. Composition of word representations improves semantic role labelling. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 407–413, Doha, Qatar. Association for Computational Linguistics.

Michel Simard. 2014. Clean data for training statistical MT: the case of MT contamination. In *Proceedings of the Eleventh Conference of the Association for Machine Translation in the Americas*, pages 69–82, Vancouver, BC, Canada.

Andreas Stolcke. 2002. SRILM – An extensible language modeling toolkit. In *Processdings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, pages 901–904.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1387–1392.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, Barry Haddow, and Ondřej Bojar. 2016. Edinburgh's statistical machine translation systems for wmt16. In *Proceedings of the First Conference on Machine Translation*, pages 399–410, Berlin, Germany. Association for Computational Linguistics.

# Alibaba Submission to the WMT18 Parallel Corpus Filtering Task

**Jun Lu, Xiaoyu Lv, Yangbin Shi, Boxing Chen**
Machine Intelligence Technology Lab, Alibaba Group
Hangzhou, China
{joelu.luj, anzhi.lxy, taiwu.syb, boxing.cbx}@alibaba-inc.com

## Abstract

This paper describes the Alibaba Machine Translation Group submissions to the WMT 2018 Shared Task on Parallel Corpus Filtering. While evaluating the quality of the parallel corpus, the three characteristics of the corpus are investigated, i.e. 1) the bilingual/translation quality, 2) the monolingual quality and 3) the corpus diversity. Both rule-based and model-based methods are adapted to score the parallel sentence pairs. The final parallel corpus filtering system is reliable, easy to build and adapt to other language pairs.

## 1 Introduction

The parallel corpus is an essential resource for machine translation and multilingual natural language processing. Apart from the quantity and domain, the quality of parallel corpus is also very important in MT system training (Koehn and Knowles, 2017; Khayrallah and Koehn, 2018). The Internet contains a large number of multilingual resources, including parallel and comparable sentences (Resnik and Smith, 2003). Many successful machine translation systems are built using the corpus crawled from the web. But in practice, this kind of parallel corpus may be very noisy. The task of Parallel Corpus Filtering tackles the problem of cleaning noisy parallel corpus.

In this task, we can divide the corpus cleaning task into three parts. Firstly, a *high-quality* parallel sentence pair should have the property that its target sentence precisely translates the source sentence, and vice versa. In this task, we attempt to quantify the translation quality (also called bilingual score) and accuracy of the sentence pair. Secondly, the quality of the target and/or source sentences of the parallel corpus should also be evaluated. In this work, the target side sentences are concerned a lot for their importance in NMT. Thirdly, as described by the *Parallel Corpus Filtering* task, the participants should not pay attention to the domain-relatedness. We need to focus on all the domains so that the resulting MT system can be widely used. So the diversity should be evaluated while subsampling the parallel corpus. Finally, the three characteristics of the parallel corpus are combined to build the final clean corpus.

The paper is structured as follows: Section 2 describes our methods which are used in parallel corpus filtering. Section 3 specifies the experiments and results. The dataset for building model-based methods is also detailed in this section. Conclusions are drawn in Section 4.

## 2 Parallel Sentence Pairs Scoring Methods

In this section, three kinds of scoring/filtering methods are detailed.

### 2.1 Bilingual Quality Evaluation

Here, we describe the noisy corpus filtering rules and two kinds of translation quality evaluation methods: (1) Word Alignment Based bilingual scoring and (2) Bitoken CNN Classifier based bilingual scoring(Chen et al., 2016).

**Rule-based Filtering**

A series of heuristic rules are applied to filter *bad* sentence pairs. They are simple but efficient, which are described below.

- The length ratio of source sentence to target sentence. Sentence length is calculated as the number of tokens/words. In our system, the ratio is set between 0.4 and 2.5.
- The edit distance between the source token sequence and the target token sequence. A

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 917–922
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64109

small edit distance indicates that the source and target sentences are very similar. This kind of corpus harms the performance of the NMT system a lot (Khayrallah and Koehn, 2018). Besides, the edit distance can be normalized by the average length of source and target sentence length, which represents the *edit distance ratio*. Both edit distance and edit distance ratio are used to filter sentence pairs in which the source and target sentence are similar. In our system, a sentences pair will be dropped if its edit distance is less than 2 or edit distance ratio is less than 0.1.

- The consistency of special tokens (Taghipour et al., 2010). For example, the high-quality sentence pairs should contain the same email address in both source and target sentences (if exists). In this task, special tokens are an email address, URL, and a big Arabic number.

### Word Alignment-based Bilingual Scoring

The word alignment model can be used for evaluating the translation quality of bilingual sentence pairs (Khadivi and Ney, 2005; Taghipour et al., 2010; Ambati, 2011). Inspired by the work of (Khadivi and Ney, 2005), we simplify the original algorithm, and the translation score of sentence pairs is given below:

$$\begin{aligned} score(s,t) =& \frac{1}{m} \sum_{s_i,t_j \in a_{s2t}} \log p(t_j|s_i) \\ &+ \frac{1}{n} \sum_{s_i,t_j \in a_{t2s}} \log p(s_i|t_j) \qquad (1) \end{aligned}$$

In Equation (1), $s$ and $t$ represent the source and target sentences respectively, $p(w_1|w_2)$ indicates the word translation probability, $a_{s2t}$ indicates the source words to target words alignment, $m$ and $n$ are the lengths of source and target sentences.

In this task, the word alignment model is trained on a clean parallel corpus provided by *WMT18 New Translation Task*. We use the fast_align toolkit (Dyer et al., 2013) to train the model, and get the forward and reverse word translation probability tables.

This model is also called alignment scoring model.

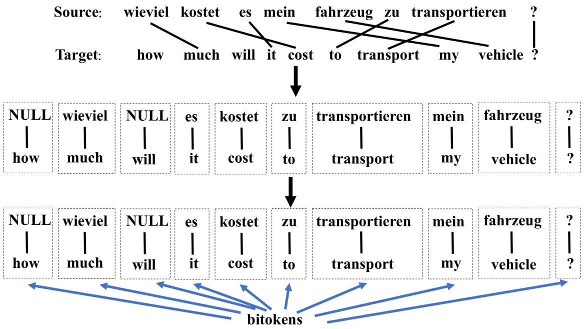

Source: wieviel kostet es mein fahrzeug zu transportieren ?
Target: how much will it cost to transport my vehicle ?
NULL how
wieviel much
NULL will
es it
kostet cost
zu to
transportieren transport
mein my
fahrzeug vehicle
? ?
bitokens

Figure 1: Bitoken sequence

### Bitoken CNN Classifier-based Bilingual Scoring

Following the work of (Chen et al., 2016), a bitoken CNN based scoring model is built for translation quality evaluation.

In this model, the *bitokens* are extracted from aligned sentence pairs. Figure 1 shows how a bitoken sequence can be obtained from a word-aligned sentence pair. Each bitoken in the sequence is treated as a word, and each bitoken sequence is treated as a normal sentence. Then these bitoken sentences are fed to the CNN Classifier to build the bilingual scoring model. For every candidate sentence pair, this model will give two probabilities: $p_{pos}$ and $p_{neg}$, and the quality score is treated as $score_{bitoken} = p_{pos} - p_{neg}$. For the train data set, the bitoken sequences obtained from the high-quality corpus are labeled as positive. As for the negative train data, we manually construct some noisy data based on the clean data.(Lample et al., 2017) For example, shuffle the target side sentences of the clean parallel corpus, or randomly delete the source or target sentence's words. So the negative bitoken sequences could be obtained from this unparallel corpus.

This scoring model can also be called bitoken_CNN scoring model.

## 2.2 Monolingual Quality Evaluation

### Rule based Filtering

A few rules are applied to filtering the sentence pairs whose source or target side are *not good*. These rules are:

- The length of the sentence which is too short ($\leq 2$ words) or too long ($> 80$ words) will be dropped.

- The ratio of valid tokens counts to the length of the sentence. Here, valid tokens are the tokens which contain the letters in the corresponding language. For example, a valid token in English should contain English letters. In our system, the sentence is filtered if its valid-tokens ratio is less than 0.2.
- Language filtering. For German-English parallel corpus, the source and target sentences' languages should be English and German. We can detect the sentence's language by using a language detection tool we developed[1]. The sentences pair is filtered if the languages of its source and target sides are not German and English.

### Language Model Scoring

We use the language model to evaluate the quality of sentences. The language model is successfully used to select domain-related corpus (Yasuda et al., 2008; Moore and Lewis, 2010). Besides, the language model can also be used to filter out ungrammatical data (Denkowski et al., 2012; Allauzen et al., 2011), which is suitable for this task.

In our corpus filtering system, we focus on the quality of target sentences, i.e. English sentences, as they are more important in NMT. Firstly, a large language model is built on all available English monolingual corpus provided by WMT18. The training corpus is cleaned using some rules mentioned above. Then the normalized-length language model score can be regarded as the monolingual quality score. But in practice, this method has a shortcoming: it gives lower scores for the good sentences that contain rare words. The training corpus needs to be generalized to overcome this shortness, for example, we can replace the words that occur less than 10 times in LM train corpus with their part of speech tag(Axelrod et al., 2015). Finally, the language model is re-built on the generalized corpus.

## 2.3 Corpus Diversity

### Rule-based Filtering

We could use a simple rule to reduce the number of similar sentence pairs. Firstly, source and target sentences should be generalized. In our experiment, for the English sentence, the generalization is done by removing all the characters except for English letters. Also, a similar operation is done for generalizing German sentences. After that, if some sentence pairs have the same generalized source or target sentences, the sentence pair that has the highest quality score will be selected.

### N-gram based Diversity Scoring

In this method, we aim to sub-select a corpus which contains a variety of N-grams. Such a corpus is regarded as high diversity. We follow the work of (Ambati, 2011; Biçici and Yuret, 2011), with the motivation for introducing a feature decay function for the n-gram weight. In our system, after selecting a subset $S_1^{j-1}$, the next sentence $s_j$'s diversity score is given by:

$$f(s_j|S_1^{j-1}) = \frac{\sum_{n=1}^{N}\sum_{ng\in NG(s_j,n)} weight(ng,j-1)}{norm(s_j)} \tag{2}$$

$$weight(ng,j-1) = Freq(ng,S) * e^{-\lambda * Freq(ng,S_1^{j-1})},$$

where $S_1^{j-1}$ represents the set of selected sentences which contains $1^{st}$ to $(j-1)^{th}$ sentences, and $S$ is the whole sentences pool to be selected.

$f(s_j|S_1^{j-1})$ is the diversity score of sentence $s_j$ under the condition that corpus $S_1^{j-1}$ is selected.

$NG(s_j,n)$ is all $n$-grams of size $n$ in sentence $s_j$. $|NG(s_j,n)|$ is the size of the $NG(s_j,n)$.

$norm(s_j)$ is the normalization factor for sentence $s_j$, and equals $\sum_{n=1}^{N}|NG(s_j,n)|$.

$Freq(ng,S)$ is the frequency of $n$-gram in selection data $S$.

$\lambda$ is the exponential decay hyper parameter, $\lambda = 1$ in our experiment.

The equation (2) indicates that the $n$-gram is weighted by its frequency in the pool set $S$ and selected set $S_1^{j-1}$. The higher the frequency of n-grams in the selected set, the lower the weight; the higher the frequency of n-gram in the pool set, the higher the weight. In practice, firstly, the sentences pairs in the pool $S$ are sorted by their quality scores(combined by bilingual and monolingual score) in descending order. Then the selection method described above is carried out on the target side of the bilingual corpus.

### Parallel Phrases Diversity Scoring

The N-gram based Diversity Scoring is commonly used for selecting monolingual sentences with high diversity. Here we aim to sub-select a bilingual corpus which contains a variety of parallel

[1] This tool is similar to Google's CLD2: https://github.com/CLD2Owners/cld2

phrases. With this kind of corpus, the MT model will learn more translation knowledge.

Firstly, we use the fast_align toolkit to train a word alignment model. And then the phrase table of the corpus can be extracted by using the Moses toolkit. Next, we can obtain the parallel phrases pairs for each sentence pair from the phrase table using the methods of maximum matching. Finally, following the method described in section *N-gram based Diversity Scoring*, the same selection procedure (in which, N-gram is replaced by phrase pairs) is used for sentence pairs' scoring. In our system, it works best when the phrase length is less than 7.

### 2.4 Methods Combination and corpus sampling

In our corpus filtering system, all the methods are combined into a pipeline.

First of all, we apply all the bilingual and monolingual rules to filter very noisy sentence pairs. Then, two bilingual scores and target side language model score could be produced by the above corresponding models. These three scores are individually normalized and then linearly combined to produce a single quality score. Here, the weights of these scores are selected with gird search method(Hsu et al., 2003). After that, we sort the sentence pairs by their corresponding quality scores in the descending order. The diversity method is then used to re-score/re-order the corpus. Finally, we select two sets of the top-N sentence pairs that contain totally 10 million words and 100 million words.

## 3 Experiments and Results

In this section, we specify the experimental settings and results in corpus filtering task.

### 3.1 Corpora and Settings

The selection data pool[2] is provided by *WMT18 Corpus Filtering Task*, which contains about 100 million sentences pairs. It is very noisy. The task's participants are asked to sub-select sentence pairs that amount to (a) 100 million words and (b) 10 million words.[3] The quality of the resulting subsets is determined by the BLEU scores of a statistical machine translation (Moses, phrase-based) and neural machine translation system (Marian) trained on this data. In our SMT and NMT experiments, we used the SMT and NMT configuration that are provided by the task organizer[4], as well as the development and test set.

While building the alignment scoring model, after using the bilingual and monolingual filtering rules, 4,337,154 sentence pairs are selected from the corpora provided by the *WMT18 English-German news translation task*. Next, the fast_align tool is used to build the word alignment model on the clean corpus, and then we can obtain the forward and reverse word translation probability tables.

When building the bitoken_CNN scoring model, 20,000 positive labeled bitoken sequences and 20,000 negative labeled bitoken sequences are constructed. The fast_align toolkit is also used here. Then, we use the *CONTEXT*[5] toolkit to train the CNN models. The bitokens' embedding vectors are trained by *word2vec*[6], and the size of each vector was set to 200.

For target sentences' quality evaluation, we use the KenLM(Heafield et al., 2013) toolkit to train the normal and generalized LM. The clean training corpus contains 60 million English sentences, which are sub-selected from the corpora provided by *WMT18 News Translation Task*.

### 3.2 Experimental Results

Firstly, the whole corpus which contains about 100 million sentence pairs was evaluated by training the SMT and NMT system. The final BLEU scores are 21.21 and 7.8 respectively. This experiment shows that the whole corpus is really noisy.

Other experimental results are detailed in Table 1. The randomly sub-selected corpus' performance is also very poor. The *sys_1* system uses the bilingual/monolingual rules and alignment scoring, which performed much better. We replace the alignment scoring method by bitoken_CNN method and then build the *sys_2* system. We find that the alignment scoring method and bitoken_CNN method are very similar in sentences pairs scoring. As a result, a lot of sentence pairs (about 70% in the subset) are selected by both methods. The two methods are combined in *sys_3*, which has a little improvement. While combining,

[2] http://www.statmt.org/wmt18/parallel-corpus-filtering-data/data.gz

[3] http://www.statmt.org/wmt18/parallel-corpus-filtering.html

[4] http://www.statmt.org/wmt18/parallel-corpus-filtering-data/dev-tools.tgz

[5] http://riejohnson.com/cnn_download.html

[6] https://code.google.com/archive/p/word2vec/

| System ID | Method | 10M words subset | | | 100M words subset | | |
|---|---|---|---|---|---|---|---|
| | | sentence pairs count ($\times 10^6$) | SMT | NMT | sentence pairs count ($\times 10^6$) | SMT | NMT |
| - | Random subset | 1.31 | 15.25 | 7.73 | 8.23 | 18.21 | 7.57 |
| sys_1 | bilingual & monolingual rules<br>+ Alignment scoring | 1.29 | 20.57 | 23.23 | 7.56 | 25.15 | 30.02 |
| sys_2 | bilingual & monolingual rules<br>+ bitoken_CNN scoring | 1.09 | 21.02 | 23.69 | 6.45 | 25.19 | 30.33 |
| sys_3 | bilingual & monolingual rules<br>+ Alignment + bitoken_CNN | 0.46 | 21.93 | 24.14 | 5.05 | 25.13 | 30.43 |
| sys_4 | sys_3 + Language Model | 0.76 | 23.53 | 25.01 | 5.41 | 25.77 | 31.44 |
| sys_5 | sys_4 + Diversity Evaluation | 0.64 | 23.79 | 25.34 | 5.41 | 25.77 | 31.44 |

Table 1: Methods used in Corpus selection and their performance

the original scores are normalized to the interval $[0, 1]$, and then the linear model is used to produce a new score. In *sys_3* system, the weights of alignment score and bitoken_CNN score are 0.4 and 0.6 respectively.

The *sys_4* introduced language mode score based on *sys_3*. The weights of the alignment score, bitoken_CNN score, and the language model score are 0.4, 0.6 and 0.8 respectively. It shows that the language model is useful in selecting clean sentences pairs.

Finally, based on *sys_4*, the corpus diversity filtering rules and scoring are introduced in *sys_5*. We find that the diversity method (only *Parallel Phrases Diversity Scoring* is used in *sys_5* system) works well in selecting the smaller subset corpus, e.g. the 10 million words corpus. For large subset corpus selection, it almost has no improvement. We attribute this to the sufficiently high diversity of larger subset corpus.

## 4 Conclusions

In this paper, we present our corpus filtering system for the *WMT 2018 Corpus Filtering Task*. In our system, sentence pairs are evaluated in three aspects: (1) the bilingual translation quality, (2) the monolingual quality of the source and target sentences and (3) the diversity of the sub-selected corpus. Our experiments show that all the methods are contributed to building a cleaner parallel corpus.

## References

Alexandre Allauzen, Hélene Bonneau-Maynard, Hai-Son Le, Aurélien Max, Guillaume Wisniewski, François Yvon, Gilles Adda, Josep M Crego, Adrien Lardilleux, Thomas Lavergne, et al. 2011. Limsi@ wmt11. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 309–315. Association for Computational Linguistics.

Vamshi Ambati. 2011. *Active learning and crowd-sourcing for machine translation in low resource scenarios*. Ph.D. thesis, University of Southern California.

Amittai Axelrod, Philip Resnik, Xiaodong He, and Mari Ostendorf. 2015. Data selection with fewer words. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 58–65, Lisbon, Portugal. Association for Computational Linguistics.

Ergun Biçici and Deniz Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283. Association for Computational Linguistics.

Boxing Chen, Roland Kuhn, George Foster, Colin Cherry, and Fei Huang. 2016. Bilingual methods for adaptive training data selection for machine translation. In *Proc. of AMTA*, pages 93–103.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The cmu-avenue french-english translation system. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 261–266. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria.

Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. 2003. A practical guide to support vector classification.

Shahram Khadivi and Hermann Ney. 2005. Automatic filtering of bilingual corpora for statistical machine translation. In *International Conference on Application of Natural Language to Information Systems*, pages 263–274. Springer.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 1–10.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224. Association for Computational Linguistics.

Philip Resnik and Noah A Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

Kaveh Taghipour, Nasim Afhami, Shahram Khadivi, and Saeed Shiry. 2010. A discriminative approach to filter out noisy sentence pairs from bilingual corpora. In *Telecommunications (IST), 2010 5th International Symposium on*, pages 537–541. IEEE.

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumita. 2008. Method of selecting training data to build a compact and efficient translation model. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

# UTFPR at WMT 2018: Minimalistic Supervised Corpora Filtering for Machine Translation

**Gustavo H. Paetzold**
Federal University of Technology - Paraná / Brazil
ghpaetzold@utfpr.edu.br

## Abstract

We present the UTFPR systems at the WMT 2018 parallel corpus filtering task. Our supervised approach discerns between good and bad translations by training classic binary classification models over an artificially produced binary classification dataset derived from a high-quality translation set, and a minimalistic set of 6 semantic distance features that rely only on easy-to-gather resources. We rank translations by their probability for the "good" label. Our results show that logistic regression pairs best with our approach, yielding more consistent results throughout the different settings evaluated.

## 1 Introduction

It is no secret that Machine Translation (MT) systems have a wide array of applications, which range from translating news to multiple languages in order to more widely spread useful information, to producing translated transcriptions of real-time audio so that people from different places can communicate more easily.

MT systems have evolved considerably throughout recent years due mainly to the widespread adoption of neural machine translation (NMT) approaches. Attention-based encoder-decoders (Bahdanau et al., 2014) and neural semantic encoders (Munkhdalai and Yu, 2016) are just some examples of recurrent neural network architectures that have achieved great success in this task.

But regardless of how much MT approaches have evolved from a modelling standpoint, both modern and legacy approaches learn from the same type of information: parallel data containing hand-crafted translations. This data usually takes the form of millions (sometimes billions) of parallel original-to-translated sentences, and are often extracted from translated versions of documents, such as news articles (Bojar et al., 2017), and subtitles (Lison and Tiedemann, 2016).

Despite being hand-crafted, sometimes these datasets contain a lot of spurious translation examples that would not necessarily teach anything useful to an MT model, potentially compromising its performance. Consequently, it is important to filter these datasets in order to maximise the model's performance. Tiedemann (2012) and Lison et al. (2018) effectively filter large parallel corpora extracted from subtitles by using unsupervised metrics that combine features such as translation probabilities, language model probabilities, etc. In this contribution, we attempt to elaborate on the ideas of Tiedemann (2012) and Lison et al. (2018) by using such features as input to supervised machine learning models.

In what follows, we present the UTFPR systems for the WMT 2018 parallel corpus filtering task: A minimalistic approach that aims at combining easy-to-harvest features with classic supervised binary classification models to create efficient translation filters.

## 2 Task Description

The WMT 2018 parallel corpus filtering task is a very simple one: given a large dataset containing many automatically harvested translations, rank them according to their quality i.e. how useful one can expect them to be to an MT system.

The dataset provided contains around 1 billion words from English-to-German translations gath-

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 923–927
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64110

ered as part of the Paracrawl project (Buck and Koehn, 2016). The translations were of mixed domain, and among them are many spurious ones, such as misaligned translations, incomplete translations, translations with non-English and/or non-German sentences, etc. Participants were allowed to use the parallel corpora[1] from the WMT 2018 MT shared task to train their systems, if they wished to do so.

Participants were tasked with creating systems that assign a quality score to each translation in the dataset. To evaluate the systems, the organizers subsampled the dataset by choosing the $N$ highest quality translations, training MT systems with them, then using traditional MT evaluation metrics to measure their performance. More details on the MT systems and evaluation metrics used are provided in Section 4.

## 3 Approach

In order to rank translations according to their quality, we've conceived a minimalistic supervised binary classification approach that relies on features that are easy to produce, and can hence be calculated even for resource-limited languages. The pipeline of our approach is illustrated in Figure 1.

First, we create a binary classification dataset using a set of high-quality English-German translations. The goal of this step is to create a very contrasting set of instances that greatly differed in terms of how coherent the source in English aligned with its German target. We create our dataset through the following steps:

1. We split the dataset in two equally sized portions, which we will henceforth refer to as "positive" and "negative" halves.

2. We then keep the positive half as it is, and shuffle the German side of the translations in the negative half, consequently misaligning the source and target side of the translations.

3. Finally, we assign label 1 (good quality) to all instances in the positive half, and -1 to the ones in the negative half (bad quality).

With our dataset at hand, we then calculate 6 features for each instance:

- The cosine distance between the average embedding vector of all content words in the source and target sentences.

- The minimum, maximum, and average cosine distance between the word embeddings of all possible word pairs in the source and target sentences.

- The proportion of words in the English source that have at least one ground-truth translation in the German target according to a dictionary.

- The proportion of words in the German target that have at least one ground-truth translation in the English source according to a dictionary.

These features have the main goal of capturing the overall semantic distance between the source and target in different ways. Notice that, since we prioritised creating an efficient and extensible approach to this task, we refrained from trying to exploit other features that attempt to capture syntactic properties, which require for parsers, which are often scarce for resource-limited languages.

To calculate our cosine distance features, we use the pre-trained 300-dimension English-German bilingual embeddings made available by the MUSE project (Lample et al., 2017). These embeddings offer a common distributional feature space for both English and German, and allow for us to calculate the cosine distance between English and German words. For the translation precision features, we used the English-German ground truth dictionary also made available by the MUSE project. These dictionaries are derived in unsupervised fashion from the same learning process that originate the previously described embeddings. Both of these resources can be obtained with raw text, without the need for parallel corpora, which makes our features easily obtainable for the great majority of languages. We treat as content words any words that are not featured in a list of stop words.

After feature calculation, we train a binary classification model over our dataset. At test time, we produce quality scores for unseen instances by calculating the same 6 features, passing them through our model, then extracting the probability of the positive class (label 1). To create a set of filtered translations, we rank the translations according to

[1] http://statmt.org/wmt18/translation-task.html

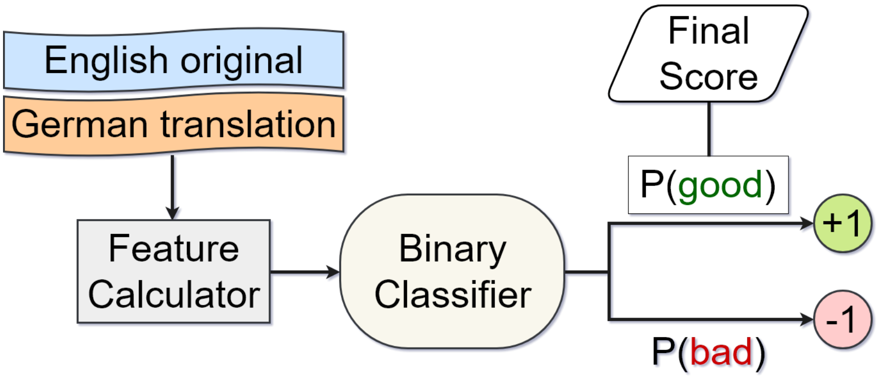


**Figure 1:** Architecture of the UTFPR systems

their positive class probabilities and choose the ones with highest scores. We name our approach UTFPR in reference to the university sponsoring this contribution.

## 4 Experimental Setup

As mentioned in Section 2, we submit our results to the parallel corpus filtering shared task of WMT 2018, of which the test set contains roughly one billion unfiltered parallel English-German translations. To train our supervised model, we use the Europarl v7 parallel corpus (Koehn, 2005), which contains $1,920,209$ translations.

For learning, we experiment with three classification models: Logistic Regression (UTFPR-LR), Decision Trees (UTFPR-DT), and Random Forests (UTFPR-RF). We chose them because they use a varying array of learning methods, and can be trained efficiently even when presented with hundreds of millions of input instances.

To evaluate our approach, the shared task organizers first created two sub-sampled sets of parallel translations containing the 10 million and 100 million highest scoring translations in the test set. They then used these sets to train both statistical (SMT) and neural MT (NMT) models using the Moses (Koehn et al., 2007) and Marian (Junczys-Dowmunt et al., 2018) toolkits, and evaluated the models according to BLEU-c (Koehn, 2011) over a combination of the newstest 2018[2], iwslt 2017[3], Acquis[4], EMEA[5], Global Voices[6], and KDE[7] datasets.

## 5 Results

We compare our approach to the 5 systems from the WMT 2018 parallel corpus filtering task with the highest and lowest average BLEU-c scores. The results illustrated in Table 1 reveal that, although our models do not fair very well against more sophisticated strategies, they do perform more consistently than other strategies of similar performance across all the settings evaluated; one can observe that the main reason why our logistic regressor outperforms the bottom five shared task systems is because it achieves similar BLEU-c scores in all settings, while the bottom five achieve unusually low BLEU-c scores in some settings (particularly 10M sentences for NMT). However, this is not necessarily a strong point of our approach, since one would expect to achieve significantly higher scores in settings where the MT systems are being fed more sentences, specially in the case of NMT. This suggests that our models may be prone to choosing redundant/repetitive content.

It can also be noted that, overall, the logistic regression model performs much better than both our decision trees and random forests, specially for NMT, where the difference between them reaches upwards of 16.08 BLEU-c points. Inspecting the highest scores produced by these models, we found that our logistic regressor and the tree-based models prioritise much different translations. Both our decision tree and random forest assign higher scores to very short translation pairs averaging 15 tokens in length on either side, while our logistic regressor prioritises much longer ones, averaging 40 tokens in length on either side. We noticed that, although the shorter translation pairs prioritised by our tree-based models often feature a slimmer array of translation errors, they seem much less use-

[2] http://statmt.org/wmt18/translation-task.html
[3] https://sites.google.com/site/iwsltevaluation2017
[4] https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis
[5] http://opus.nlpl.eu/EMEA.php
[6] http://opus.nlpl.eu/GlobalVoices.php
[7] http://opus.nlpl.eu/KDEdoc.php

|  | SMT |  | NMT |  |  |
|---|---|---|---|---|---|
|  | **10M** | **100M** | **10M** | **100M** | **Average** |
| Microsoft | 24.45 | 26.50 | 28.62 | 32.06 | 27.91 |
| RWTH | 24.58 | 26.21 | 28.01 | 31.29 | 27.52 |
| Alibaba | 24.11 | 26.44 | 27.60 | 31.93 | 27.52 |
| Alibaba-Div | 24.11 | 26.42 | 27.60 | 31.92 | 27.51 |
| NRC | 23.89 | 26.40 | 27.41 | 31.88 | 27.39 |
| **UTFPR-LR** | **20.81** | **22.35** | **21.75** | **22.23** | **21.79** |
| **UTFPR-DT** | 17.55 | 20.67 | 11.44 | 11.88 | 15.38 |
| **UTFPR-RF** | 13.22 | 16.96 | 6.57 | 6.15 | 10.72 |
| AFRL-Small | 21.93 | 22.89 | 13.49 | 21.05 | 19.84 |
| DCU-System 4 | 15.67 | 21.19 | 6.27 | 18.60 | 15.43 |
| DCU-System 3 | 15.26 | 21.09 | 5.01 | 18.39 | 14.94 |
| DCU-System 2 | 12.86 | 18.57 | 3.42 | 8.61 | 10.86 |
| DCU-System 1 | 6.56 | 13.22 | 3.34 | 4.78 | 6.98 |

**Table 1:** Parallel corpus filtering results with respect to the average BLEU-c scores obtained over the datasets described in Section 4. The first and last five lines feature, respectively, the five systems that achieved the highest and lowest average BLEU-c scores in the task. Boldface numbers highlight the highest BLEU-c scores achieved among the UTFPR systems.

ful to an MT system. Most of them are translations of dates, article titles, ads, and list items, which we expect would offer little to no insight on how to translate longer, more elaborate sentences. In contrast, the longer translations prioritised by our logistic regressor feature more meaningful, complex sentences, which is most likely why they make for better input to MT models.

## 6 Conclusions

In this contribution, we presented the UTFPR systems submitted to the WMT 2018 parallel corpus filtering task. Our supervised systems discern between good and bad translations using classic binary classification models, and use as input a minimalistic set of 6 features that aim to capture the semantic distance between original and translated sentences without relying neither on syntactic information or scarce resources and tools.

We found that our approach performs best when employing logistic regression. Overall, our best performing system places 41th, when considering the BLEU-c average of all outcomes evaluated. In the future, we aim to evaluate the effectiveness of applying more elaborate dataset creation methods for training that produce more types of errors, employing more sophisticated neural models for the task, and incorporating cost-effective syntactic clues into the feature set.

## 7 Acknowledgments

We would like to thank the Federal University of Technology - Paraná for supporting this contribution.

## References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the 2nd Conference on Machine Translation*, pages 169–214.

Buck, Christian and Philipp Koehn. 2016. Findings of the wmt 2016 bilingual document alignment shared task. In *Proceedings of the 1st Conference on Machine Translation*, pages 554–563.

Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of the 56th annual meeting of the ACL*, pages 116–121.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL*, pages 177–180.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Koehn, Philipp. 2011. What is a better translation? reflections on six years of running evaluation campaigns. *Tralogy 2011*, page 9.

Lample, Guillaume, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Lison, Pierre and Jrg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the 10th LREC*.

Lison, Pierre, Jrg Tiedemann, and Milen Kouylekov. 2018. Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the 11th LREC*.

Munkhdalai, Tsendsuren and Hong Yu. 2016. Neural semantic encoders. *CoRR*, abs/1607.04315.

Tiedemann, Jrg. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the 8th LREC*.

# The ILSP/ARC submission to the WMT 2018 Parallel Corpus Filtering Shared Task

**Vassilis Papavassiliou** **Sokratis Sofianopoulos**

**Prokopis Prokopidis** **Stelios Piperidis**

Institute for Language and Speech Processing/Athena RC
Athens, Greece
{vpapa, s_sofian, prokopis, spip}@ilsp.gr

## Abstract

This paper describes the submission of the Institute for Language and Speech Processing/Athena Research and Innovation Center (ILSP/ARC) for the WMT 2018 Parallel Corpus Filtering shared task. We explore several properties of sentences and sentence pairs that our system explored in the context of the task with the purpose of clustering sentence pairs according to their appropriateness in training MT systems. We also discuss alternative methods for ranking the sentence pairs of the most appropriate clusters with the aim of generating the two datasets (of 10 and 100 million words as required in the task) that were evaluated. By summarizing the results of several experiments that were carried out by the organizers during the evaluation phase, our submission achieved an average BLEU score of 26.41, even though it does not make use of any language-specific resources like bilingual lexica, monolingual corpora, or MT output, while the average score of the best participant system was 27.91.

## 1 Introduction

There is a growing literature on using web-acquired data for constructing various types of language resources, including monolingual and parallel corpora. As shown in, among others, Pecina et al. (2014) and Rubino et al. (2015), such resources can be exploited in training generic or domain-specific machine translation systems. Nevertheless, compared to the acquisition of monolingual data from the web, construction of parallel resources is more challenging. Apart from the identification of document pairs that are translations of each other and can be crawled from multilingual websites, the extraction of sentence pairs and, crucially, the selection of sentence pairs of good quality are far from straightforward.

Zariņa et al. (2015) exploit already available parallel corpora in order to get word alignments, which are then used to identify mistranslations. Denkowski et al. (2012) use N-gram language models built from monolingual corpora to estimate probabilities of source and target sentences, in a manner of assigning high scores to grammatical sentences and lower scores to ungrammatical sentences and non-sentences such as site maps, large lists of names, and blog comments. Aiming to select sentence pairs of good adequacy and fluency, Xu and Koehn (2017) generate probabilistic dictionaries and n-gram models from Europarl corpora. Taghipour et al. (2011) and Cui et al. (2013) extract features based on translation and language models, and word alignments from the dataset under examination (i.e. this dataset is used to train models instead of using external language resources) and then apply unsupervised techniques such as outlier detection of estimated probability density and graph-based random walk algorithm to discard sentence pairs that are of limited or no importance. In the case of web acquired data, shallow features like aligners' scores, length ratio, and patterns in URLs from which the content was originated, have been proposed (Esplà-Gomis and Forcada, 2010).

In a different manner, many researchers have approached data selection as a domain-matching issue. For instance, Duh et al. (2013) proposed the use of a neural language model trained on a domain-specific corpus to identify in-domain sentence pairs in a large corpus.

This paper describes the submission of ILSP/ARC for the WMT 2018 Parallel Corpus Filtering shared task. The task consisted in cleaning a very noisy English-German parallel corpus of 104 million sentence pairs provided by the organizers, with each EN-DE sentence pair accompanied by a score generated by the Hunalign sentence aligner.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 928–933
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64111

The participants were to assign a quality score for each sentence pair, with higher scores indicating sentence pairs of better quality. As reported in the shared task webpage[1], "Evaluation of the quality scores will be done by subsampling 10m and 100m [EN] word corpora based on these scores, training statistical and neural machine translation systems with these corpora, and evaluating translation quality on blind test sets using the BLEU score." Given that the organizers discouraged participants from subsampling the corpus for relevance to a specific domain (e.g. the news domain), domain adaptation approaches like the ones mentioned above seem to not fit this task.

In the shared task webpage, the organizers also released a development environment with configuration files and scripts that allowed participants to subsample corpora based on quality scores and to replicate the testing procedure with a development test set.

## 2 System architecture

Our submission system is based on the cleaning module of the ILSP Focused Crawler (Papavassiliou et al., 2013), an open-source toolkit[2] that integrates all necessary software[3] for the creation of high-precision parallel resources from the web in a language-independent fashion.

The toolkit and its cleaning module have been used in research projects like the European Language Resource Coordination for the acquisition of high-precision parallel language resources (Papavassiliou et al., 2018).

### 2.1 Noise in Web acquired parallel corpora

In a pipeline for the construction of parallel corpora from the web, shortcomings of each processing step may introduce errors, usually called "noise", that affect the quality of the final output. In this shared task, the data collection pipeline of the Paracrawl[4] project was adopted for the construction of the input (i.e. the raw, very noisy parallel corpus).

Many types of noise occur due to misses during parsing HTML pages and extracting their textual content. Such errors are typically introduced when HTML code is considered text and/or page encoding is not successfully detected. Moreover, inaccurate identification of paragraph limits may lead to wrong sentence splitting and, eventually, in the alignment of incomplete sentences. False negatives in the detection of boilerplate text (i.e. navigation headers, disclaimers, etc.) may result in large numbers of (near-)duplicate sentence pairs, which are of only limited or no use for the production of good-quality language resources.

Other errors concern the accuracy of the language identification process. Even when the language of a web page is correctly detected at document level, it is possible that small parts of the page are written in another language. Thus, ignoring language detection at paragraph or sentence level may lead to sentence pairs with the wrong language in the source and/or the target side. Finally, misalignments at document and/or sentence level generate sentence pairs that are not translations of each other.

### 2.2 Filter-based clustering

Given that the existence of the types of noise discussed above is not strongly influenced by the targeted language pair, we developed a language agnostic method with the purpose of clustering sentence pairs in respect of their quality, i.e. of their correctness and usefulness for training MT engines.

The first cluster, $C_0$, includes obviously noisy sentence pairs. We assign to these pairs a 0 score in order to prohibit their participation in the subsamples to be used for training. Sentence pairs in $C_0$ match one of the following patterns:

1. sentence pairs with too short or too long EN or DE sentences (after tokenization) that would have been excluded from the training phase according to the shared task configuration. By enforcing a sentence length between 1 and 80 tokens, and a sentence length ratio less than 9 tokens (i.e. by using the default values of the Moses SMT toolkit for cleaning a corpus before training an MT system), we remove 3.42% of the sentence pairs in the input corpus. Our intuition is that most of these sentence pairs are the result of wrong HTML parsing or encoding detection.

2. sentence pairs with an EN or DE sentence that does not contain any letter in the range

[1] http://www.statmt.org/wmt18/parallel-corpus-filtering.html

[2] http://nlp.ilsp.gr/ilsp-fc/

[3] Including modules for metadata extraction, language identification, boilerplate removal, document clean-up, text classification and sentence alignment

[4] https://paracrawl.eu/releases.html

of Unicode character sets relevant to Latin scripts. This pattern discards sentence pairs (11.12% of the input corpus) that are either the result of wrong encoding detection, or contain only dates, prices, flight numbers, dimensions, products' IDs, etc.

3. sentence pairs with identical text in both languages (after removing non-Latin characters mentioned above). These sentence pairs (9.94% of the input corpus) mainly contain boilerplate elements, dates, locations, etc.[5]

4. sentence pairs for which the EN or DE parts were not in the proper language as detected by the Cybozu language detection library.[6] Sentences in these pairs (13.01% of the corpus) were often French or Spanish. As with most language detectors, the accuracy of the tool is lower during the examination of short sentences.

5. sentence pairs (1.71% of the corpus) with unusual features (e.g. words with transitions from lowercase to uppercase and vice versa, consecutive identical letters, long sequences of very short words, etc.)

6. sentence pairs consisting mostly of URLs, and emails (1.42% of the corpus)

Table 1 provides examples of sentences grouped into $C_0$ by some of the criteria described above.

In the next step of our language agnostic approach, we clustered the remaining sentence pairs using shallow features that are likely to be related to correctness of sentence alignment. Specifically, we compared the sequences of digits and symbols (e.g. punctuation marks, % , $, etc.) on each side of the remaining sentence pairs. Depending on the results (i.e. same/different digits and same/different symbols), the following four clusters, ordered from worst to best, were constructed:

$\mathbf{C_1}$ Different digits and different symbols

$\mathbf{C_2}$ Different digits and same symbols

$\mathbf{C_3}$ Same digits and different symbols

$\mathbf{C_4}$ Same digits and same symbols

Table 2 contains examples of sentence pairs grouped into clusters according to this approach.

In a final step we focused on the identification of (near) duplicates. In more detail, we normalized sentence pairs by lowercasing and removing non-Latin characters, and we examined if a sentence pair was identical to or was included in another sentence pair. When a duplicate was detected, we kept the sentence pair that belonged to a better cluster. If both sentence pairs belonged to the same cluster, we kept the longer one in terms of tokens.

By assigning the corresponding cluster number to each sentence pair as a score (i.e. 4 to pairs of $C_4$, 3 to pairs of $C_3$, etc), the sentence pairs in the provided noisy corpus were roughly ranked. We then ran the subsampling algorithm that was provided by the organizers in order to obtain the two datasets required from each participant. We noticed that the sizes of the resulting corpora exceeded the 10M and 100M EN word thresholds. This is explained by the fact that we provided only 5 scores (as many as the clusters) and the algorithm selects all sentence pairs for a score (staring from the highest) iteratively until the size of the selected subcorpus reaches the threshold. For instance, clusters $C_4$, $C_3$ and $C_2$ (i.e. sentence pairs with scores 4, 3 and 2 respectively) including more than 14M English words, were sampled for the 10M corpus! To overcome this shortcoming, in our final rankings each cluster is initially assigned to an integer of different scale (e.g. $C_1$ to score 10, $C_2$ to score 1000, etc). The score of each sentence pair is then calculated by adding the Hunalign score to the initial cluster score, with the purpose of ensuring the granularity of the scores and of keeping clusters well-separated. This ranking led to corpora of 626K and 5.7M pairs for the 10M and the 100M corpora, respectively.

For a submission based on an alternative ranking, we add the character length of each pair to the initial cluster score. Compared to the Hunalign-based scoring, this variant favors long sentences and thus results in significantly smaller corpora in terms of sentence pairs (221K pairs and 5.4M pairs for the 10M and 100M corpora, respectively).

## 3 Evaluation Results

In the evaluation experiments conducted by the organizers, four different translation systems were trained, namely (a) a Moses statistical system

[5] In future work we plan to reconsider the usefulness of this pattern in preparing parallel corpora for NMT engines.

[6] http://code.google.com/p/language-detection/

|  | EN | DE | Aligner score |
|---|---|---|---|
| 1 | Relatively extreme values are also taken into account. | Relatively extreme values are also taken into account. | 2.4 |
| 2 | www.gamersglobal.de about Risen 2 | www.gamersglobal.de ber Risen 2 | 6.3 |
| 3 | wie gehts denn so? | wo hast deins denn her? | 1.12381 |
| 4 | 5103 Dec 5104 JanFebMarAprMayJun-JulAug | 4574 FebMAprMaiJunJulAugSepOkt | 1.46471 |
| 5 | Abstr. Appl. Anal. 2014, Art. ID 363925, 7 pp. 54H25 (45G10) | Fluct. Noise Lett. 5 (2005), no. 2, L275 L282. 82C31 | 1.26739 |

Table 1: Examples of sentence pairs grouped into $C_0$ by filters focusing on sentence pairs with 1) identical text in both languages 2) sentences consisting mainly of URLs/emails/dates 3) the sentence in the first/second column detected as non-EN/non-DE, respectively 4) unusual patterns like mixture of upper- and lowercase ; 5) long sequences of short words.

(Koehn et al., 2007) trained on the 10M EN word parallel corpus, (b) a Moses system trained on the 100M EN word parallel corpus, (c) a Marian neural translation system (Junczys-Dowmunt et al., 2018) trained on the 10M EN word parallel corpus and (d) a Marian system trained on the 100M EN word parallel corpus. For all systems the official WMT 2017 news translation test set was used as a development set. According to the shared task's settings, the quality of the machine translation system is measured by BLEU score (Papineni et al., 2002) on the (a) official WMT 2018 news translation test set and (b) another undisclosed test set, which is the union of 5 test sets listed in Tables 3 and 4.

Table 3 summarizes the evaluation scores obtained using the ranking based on the combination of clusters and Hunalign scores on the various test sets. Our submission had an average BLEU score of 26.41 on the different test configurations (4 systems evaluated over 6 test sets), while the average score of the best participant system was 27.91.

It can be seen that for all datasets the best results are obtained by the NMT systems over their equivalent SMT ones, with the top one being the NMT trained over the 100M English token German-English filtered corpus. For both the Moses SMT and the Marian NMT systems there is a significant increase of the BLEU score when increasing the size of the training corpus from 10M to 100M English tokens. Specifically, for the Moses system the average increase is 16.6%, while for Marian the average increase is 21.5%.

Similarly, Table 4 lists the evaluation scores obtained with the alternative ranking scheme using the sentence length information. This submission had an average BLEU score of 24.98 on the different test configurations. Again, the best results are obtained with the NMT system trained over the 100M corpus. When comparing the average BLEU scores between the 10M and 100M systems, the SMT system shows an increase of 15.2%, while the NMT system shows a huge increase of 58.5%. Interestingly, the performance of the NMT system trained on the 10M corpus is lower than that of the SMT one. This can be attributed to the fact that the 10M corpus comprises 221K long sentence pairs, a relatively small number of sentences for NMT systems, which evaluate fluency over entire sentences. The equivalent SMT system is rather unaffected, presumably because SMT systems are based on n-gram models.

By comparing the results of the two alternative ranking schemes, we conclude that their performances are similar for the 100M corpora. This is explained by the fact that their intersection is extremely high: 5.2M sentence pairs are included in the 5.7M and 5.4M sentence pairs selected with the two schemes. Regarding the 10M corpora which differ significantly in number of sentence pairs (626K vs 221K), the performance of both schemes is similar for the SMT systems but differs for the NMT ones. In future work, we plan to carry out experiments that will provide evidence of how size and length of sentence pairs in a training corpus affect the performance of an NMT system.

## 4 Conclusions

In this paper we described the ILSP/ARC submission to the WMT 2018 Parallel Corpus Filtering

| Cluster | EN | DE | Aligner score |
|---|---|---|---|
| $C_2$ | We offer 2 comfortable bedrooms, sleeping up to 4 guests, a cot | Zwei komfortable Schlafzimmer für bis zu 4 Personen, Kinderbett | 0.41805 |
| $C_2$ | The table now has 2 columns for the 2 euro commemorative coins, because some countries will issue two different 2 euro special coins. A description can be viewed by holding the mouse over the i-symbol for a while. | Es gibt in der Tabelle 2 Spalten für 2 Euro Gedenkmnzen, da seit 2007 einige Länder mehrere 2 Euro Sondermünzen ausgeben. Über das i-Symbol kann die entsprechende Bezeichnung der Münzen angezeigt werden. | 0.49576 |
| $C_3$ | Our club for runners who have finished in Düsseldorf 10 times. We would like to honour this accomplishment. | Unser Club für alle Läufer, die bereits 10 Mal in Düsseldorf gefinished haben. Diese besondere Leistung, möchten wir auch besonders würdigen. | 1.5466 |
| $C_4$ | Austrian declaration of principles at the Conference on Security and Cooperation in Europe (Helsinki, December 1972) | Grundsatzerklärung Österreichs auf der Konferenz über Sicherheit und Zusammenarbeit in Europa (Helsinki, Dezember 1972) | 3.9431 |
| $C_4$ | A current application: The turbine sheets of the new Airbus A 380 were manufactured by a milling machine equipped by a self carrying product of WeBe Electronic GmbH. | Eine aktuelle Applikation: Die Turbinenblätter des neuen Airbusses A 380 von einer mit einem selbsttragenden WeBe-Produkt ausgerüsteten Fräsmaschine gefertigt. | 2.6620 |

Table 2: Examples of sentence pairs grouped to different clusters based on the shallow features detailed in Section 2.2.

| | SMT 10M | SMT 100M | NMT 10M | NMT 100M |
|---|---|---|---|---|
| **news2017** | 20.49 | 25.28 | 26.09 | 31.46 |
| **news2018** | 26.30 | 30.58 | 31.32 | 38.99 |
| **iwslt2017** | 18.83 | 22.82 | 21.20 | 26.57 |
| **Acquis** | 18.71 | 22.27 | 22.94 | 27.63 |
| **EMEA** | 26.50 | 30.88 | 30.17 | 35.96 |
| **GlobalVoices** | 20.20 | 23.43 | 23.39 | 28.20 |
| **KDE** | 23.78 | 26.74 | 25.73 | 30.63 |
| **average** | 22.39 | 26.12 | 25.79 | 31.33 |

Table 3: BLEU evaluation scores (ranking was based on the combination of clusters and Hunalign scores)

| | SMT 10M | SMT 100M | NMT 10M | NMT 100M |
|---|---|---|---|---|
| **news2017** | 20.82 | 25.50 | 16.32 | 31.38 |
| **news2018** | 26.91 | 30.80 | 20.33 | 39.01 |
| **iwslt2017** | 18.91 | 22.70 | 11.40 | 26.60 |
| **Acquis** | 19.34 | 22.35 | 21.13 | 27.82 |
| **EMEA** | 27.24 | 30.86 | 27.43 | 35.89 |
| **GlobalVoices** | 20.38 | 23.49 | 14.67 | 28.32 |
| **KDE** | 23.32 | 26.59 | 23.68 | 30.37 |
| **average** | 22.68 | 26.13 | 19.77 | 31.34 |

Table 4: BLEU evaluation scores (ranking was based on the combination of clusters' scores and sentences' length)

Shared Task. We explored shallow features of sentences and sentence pairs and grouped the task data in 5 clusters according to their presumed usefulness for training MT systems. Our language-pair independent submissions were not based on MT output or bilingual lexica, i.e. on resources which are often scarce or simply not available for many language pairs. Nevertheless, the results obtained from the systems trained on our submissions indicate that this language-pair independent approach yields datasets on which competitive MT systems can be built.

## Acknowledgments

This work has been supported by SMART 2015/1091 LOT 3 & LOT 2, a service contract that

implements the acquisition of language resources for the EC's Connecting Europe Facility (CEF) eTranslation platform.

## References

Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li, and Ming Zhou. 2013. Bilingual data cleaning for SMT using graph-based random walk. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 340–345, Sofia, Bulgaria. Association for Computational Linguistics.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The CMU-Avenue French-English Translation System. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 261–266, Montréal, Canada. Association for Computational Linguistics.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 678–683. Association for Computational Linguistics.

Miquel Esplà-Gomis and Mikel L. Forcada. 2010. Combining content-based and url-based heuristics to harvest aligned bitexts from multilingual sites with bitextor. *The Prague Bulletin of Mathemathical Lingustics*, 93:77–86.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vassilis Papavassiliou, Prokopis Prokopidis, and Stelios Piperidis. 2018. Discovering parallel language resources for training MT engines. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Vassilis Papavassiliou, Prokopis Prokopidis, and Gregor Thurmair. 2013. A modular open-source focused crawler for mining monolingual and bilingual corpora from the web. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 43–51, Sofia, Bulgaria. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pavel Pecina, Antonio Toral, Vassilis Papavassiliou, Prokopis Prokopidis, Aleš Tamchyna, Andy Way, and Josef Genabith. 2014. Domain adaptation of statistical machine translation with domain-focused web crawling. *Language Resources and Evaluation*, 49(1):147–193.

Raphael Rubino, Tommi Pirinen, Miquel Esplà-Gomis, Nikola Ljubešić, Sergio Ortiz Rojas, Vassilis Papavassiliou, Prokopis Prokopidis, and Antonio Toral. 2015. Abu-MaTran at WMT 2015 Translation Task: Morphological Segmentation and Web Crawling. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 184–191, Lisbon, Portugal. Association for Computational Linguistics.

Kaveh Taghipour, Shahram Khadivi, and Jia Xu. 2011. Parallel corpus refinement as an outlier detection algorithm. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 414–421.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy webcrawled parallel corpora. In *EMNLP*, pages 2945–2950. Association for Computational Linguistics.

Ieva Zariņa, Pēteris Ņikiforovs, and Raivis Skadiņš. 2015. Word alignment based parallel corpora evaluation and cleaning using machine learning techniques. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, pages 185–192, Antalya, Turkey.

# SYSTRAN Participation to the WMT2018 Shared Task on Parallel Corpus Filtering

**MinhQuang Pham, Josep Crego, Jean Senellart**
SYSTRAN / 5 rue Feydeau, Paris (France)
FirstName.LastName@systrangroup.com

## Abstract

This paper describes the participation of SYSTRAN to the shared task on parallel corpus filtering at the Third Conference on Machine Translation (WMT 2018). We participate for the first time using a neural sentence similarity classifier which aims at predicting the relatedness of sentence pairs in a multilingual context. The paper describes the main characteristics of our approach and discusses the results obtained on the data sets published for the shared task.

## 1 Introduction

Corpus-based approaches to machine translation rely on the availability and quality of parallel corpora. In the case of neural machine translation, a large neural network is trained to maximise the translation performance on a given parallel corpus. Therefore, the quality of an MT engine is heavily dependent upon the amount and quality of the training parallel sentences. Such resource is not naturally existing, and because of the process necessary to compile a parallel corpus, it may contain multiple sentence pairs that are often not as parallel as one might assume.

The primary objective of our approach is to assess whether we are able to identify parallel sentences using a flexible method that relies on deep learning architectures. Thus, eliminating the need for any domain specific feature engineering. We evaluate the feasibility of a model learnt over the same noisy data that must be cleaned. Using as few external tools as possible.

Hence, we tackle the filtering problem by means of a neural sentence similarity network, which aims at predicting the relatedness of sentence pairs. Pairs are selected according to their similarity score, thus filtering those sentences which are less likely to be translations of each other. The rest of this paper is organised as follows. After describing the filtering task we outline our similarity classifier. Next, we present experiments and results of the shared task. Finally, we draw some conclusions.

## 2 Task description

In the context of the third conference on machine translation (WMT18), the parallel corpus filtering shared task[1] tackles the problem of cleaning noisy parallel corpora. Given a noisy parallel corpus (crawled from the web), participants develop methods to filter it to a smaller size of high quality sentence pairs. Specifically, the organisers provide a very noisy 1 billion word (English token count) German-English corpus crawled from the web as part of the Paracrawl project[2]. Participants must subselect sentence pairs that amount to (a) 100 million words, and (b) 10 million words. The quality of the resulting subsets is determined by the quality of a statstical and a neural machine translation system trained on this data. The quality of the machine translation system is measured by BLEU score on the (a) official WMT 2018 news translation test and (b) another undisclosed test set.

The organisers explicit that the task addresses the challenge of *data quality* and *not domain-relatedness* of the data for a particular use case. Hence, they discourage participants from subsampling the corpus for relevance to the news domain despite being one of the evaluation test sets. Organisers thus place more emphasis on the second undisclosed test set, although they report both scores. The provided raw parallel corpus is the outcome of a processing pipeline that aimed for high recall at the cost of precision, which makes it extremely noisy. The corpus exhibits noise of all kinds (wrong language in source and target, sentence pairs that are not translations, bad language,

[1] http://www.statmt.org/wmt18/parallel-corpus-filtering.html

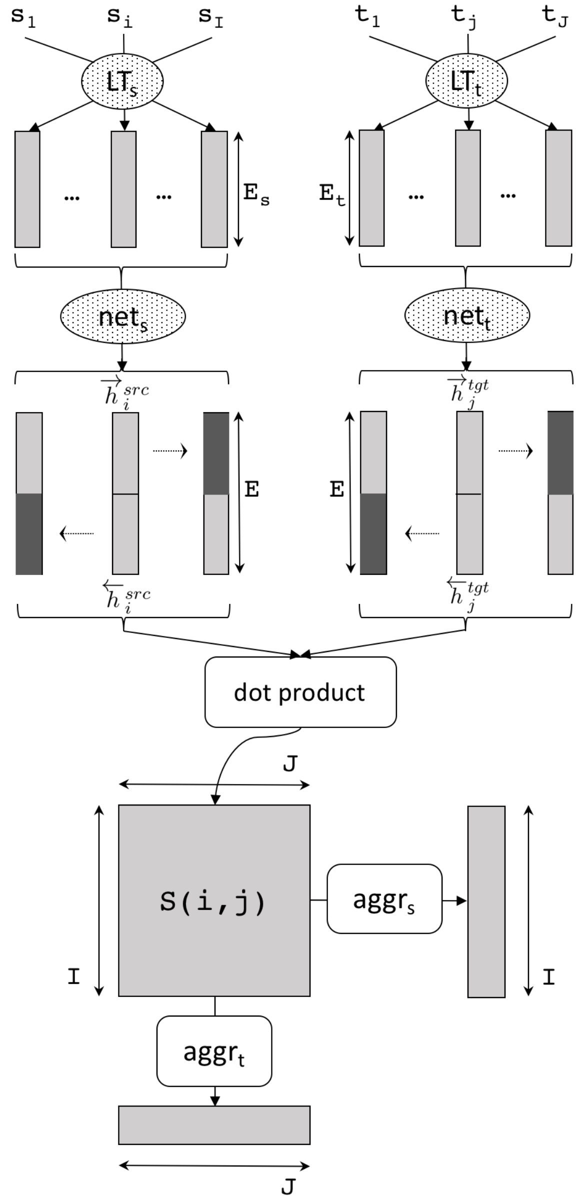
[2] https://paracrawl.eu/

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 934–938
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64112

incomplete or bad translations, etc.).

## 3 Neural Similarity Classifier

Our network architecture is very much inspired by the work on Word Alignment in (Legrand et al., 2016). Figure 1 illustrates the network. In the following, we consider a source-target sentence pair $(s, t)$ with $s = (s_1, ..., s_I)$ and $t = (t_1, ..., t_J)$.



$s_1$
$s_i$
$s_I$
$t_1$
$t_j$
$t_J$
$LT_s$
$LT_t$
$E_s$
$E_t$
$net_s$
$net_t$
$\overrightarrow{h}_i^{src}$
$\overrightarrow{h}_j^{tgt}$
E
E
$\overleftarrow{h}_i^{src}$
$\overleftarrow{h}_j^{tgt}$
dot product
J
S(i,j)
$aggr_s$
I
I
$aggr_t$
J

**Figure 1:** Illustration of the model. The network is composed of source and target word embedding lookup tables ($LT_s$ and $LT_t$) and two identical subnetworks ($net_s$ and $net_t$) that compute in context representations of source ($s_i$) and target words ($t_j$).

The model is composed of 2 Bi-directional LSTM subnetworks, $net_s$ and $net_t$, which respectively encode source and target sentences. Since both $net_s$ and $net_t$ take the same form we describe only the source architecture. The source-sentence Bi-LSTM network outputs forward and backward hidden states, $\overrightarrow{h}_i^{src}$ and $\overleftarrow{h}_i^{src}$, which are then concatenated into a single vector encoding the $i^{th}$ word of the source sentence, $h_i^{src} = [\overrightarrow{h}_i^{src}; \overleftarrow{h}_i^{src}]$. In addition, the last forward/backward hidden states (outlined using dark grey in Figure 1) are also concatenated into a single vector to represent whole sentences $h_{src} = [\overrightarrow{h}_I^{src}; \overleftarrow{h}_1^{src}]$. At this point a measure of similarity between sentences can be obtained by cosine similarity:

$$sim(h_{src}, h_{tgt}) = \frac{h_{src} \cdot h_{tgt}}{||h_{src}|| * ||h_{tgt}||} \quad (1)$$

where two vectors (embeddings) with the same orientation have a cosine similarity of 1, while two vectors with opposed orientation have a similarity of $-1$, independent of their magnitude.

Similar to (Legrand et al., 2016) our model extracts context information from source and target sentences and then computes simple dot-products to estimate word alignments. The objective function is computed at the level of words. To enable unsupervised training, we use an aggregation operation that summarizes the alignment scores for a given target word. A soft-margin objective increases scores for true target words while decreasing scores for target words that are not present. The aggregation function combines the scores of all source (or target) words for a particular target (or source) word and promotes source words which are likely to be aligned with a given target word according to the knowledge the model has learned so far. Alignment scores $S(i, j)$ are given by the dot-product $S(i, j) = h_i^{src} \cdot h_j^{tgt}$, while aggregation functions are defined as:

$$\begin{aligned} aggr_s(i, S) &= \frac{1}{r} log \left( \sum_{j=1}^{J} e^{r * S(i,j)} \right) \\ aggr_t(j, S) &= \frac{1}{r} log \left( \sum_{i=1}^{I} e^{r * S(i,j)} \right) \end{aligned} \quad (2)$$

The loss function is defined as:

$$\begin{aligned} &\mathcal{L}(src, tgt) = \\ &\qquad \sum_{i=1}^{I} log \left( 1 + e^{aggr_s(i,S) * \mathcal{Y}_i^{src}} \right) + \\ &\qquad + \sum_{j=1}^{J} log \left( 1 + e^{aggr_t(j,S) * \mathcal{Y}_j^{tgt}} \right) \end{aligned} \quad (3)$$

where $\mathcal{Y}_i^{src}$ and $\mathcal{Y}_j^{tgt}$ are vectors with reference labels containing $-1$ when the word is present in the translated sentence, and $+1$ for divergent (unpaired) words.

Further details on the network can be found in (Pham et al., 2018).

### 3.1 Training with Negative Examples

Training is performed by minimising Equation 3, for which examples with annotations for source $\mathcal{Y}_i^{src}$ and target $\mathcal{Y}_j^{tgt}$ words are needed.

As positive examples we use **paired** sentences of the parallel corpus. In this case, all words in both sentences are labelled as parallel, $\mathcal{Y}_i^{src} = -1$ and $\mathcal{Y}_j^{tgt} = -1$.

As negative examples we use random **unpaired** sentences. In this case, all words are labelled as divergent, $\mathcal{Y}_i^{src} = +1$ and $\mathcal{Y}_j^{tgt} = +1$.

In order to be able to predict less obvious divergences we **replace** random sequences of words on either side of the sentence pair by a sequence of words with the same part-of-speeches. The rationale behind this method is to keep the new sentences as grammatical as possible. Otherwise, to predict divergence the network can learn to detect non-grammatical sentences. Words that are not replaced are considered parallel ($-1$) while those replaced are assigned the divergent label ($+1$). Words aligned to some replaced words are also assigned the divergent label ($+1$).

Finally, motivated by sentence segmentation errors observed in many corpora, we also build negative examples by **inserting** a second sentence at the beginning (or end) of the source (or target) sentence pair. Words in the original sentence pair are assigned the parallel label ($-1$) while the new words inserted are considered divergent ($+1$).

In order to avoid that negative examples are easily predicted just by looking at the difference in length of training sentences we constraint all negative examples to have a difference in length not exceeding $2.0$. Very short sentences, of up to $4$ words, are accepted if the length ratio does not exceeds $3.0$.

## 4 Experiments

### 4.1 Neural Similarity Classifier

All data is preprocessed with `OpenNMT`[3], performing minimal tokenisation, basically splitting-off punctuation. After tokenisation, the $50,000$ most frequent words of each language are used as vocabulary. Each out-of-vocabulary word is mapped to a special UNK token. Word embeddings ($LT_s$ and $LT_t$) are initialised using `fastText`[4], further aligned by means of `MUSE`[5] following the unsupervised method detailed in (Lample et al., 2018). Size of embeddings is $E_s = E_t = 256$ cells. Both Bi-LSTM use 256-dimensional hidden representations ($E = 512$). We use $r = 1.0$. Optimisation of the parameters is done using the stochastic gradient descent method along with gradient clipping (rescaling gradients whose norm exceeds a threshold) to avoid the exploding gradients problem (Pascanu et al., 2013). For each epoch we randomly select $1$ million sentence pairs that we place in batches of $32$ examples. Word alignments and English part-of-speeches used to build negative examples were performed by `fast_align`[6] and `FreeLing`[7] respectively. We run $10$ epochs and start decaying at each epoch by $0.8$ when score on validation set increases. Similarity is always computed following equation 1.

### 4.2 Simple Filtering

The Corpora of the shared task contains $1$ billion word (English token count) German-English corpus crawled from the web as part of the Paracrawl project. Observing that many sentence pairs could be easily filtered out by simple rules imposed on length and language, we use a very simple filter which removes $80\%$ of the sentence pairs. Our basic filterig consists of:

- Language Identification on source and target sentences,
- removing pairs whose source-target or target-sources length ratio is higher than $6$,
- removing pairs whose source or targets length is higher than $100$.

After this simple filtering, our corpus is reduced to $22$ million sentence pairs.

## 5 Results

Participants in the shared task have to submit a file with quality scores, one per line, corresponding to

[3] http://opennmt.net

[4] https://github.com/facebookresearch/fastText

[5] https://github.com/facebookresearch/MUSE

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

[7] https://github.com/TALP-UPC/FreeLing.git

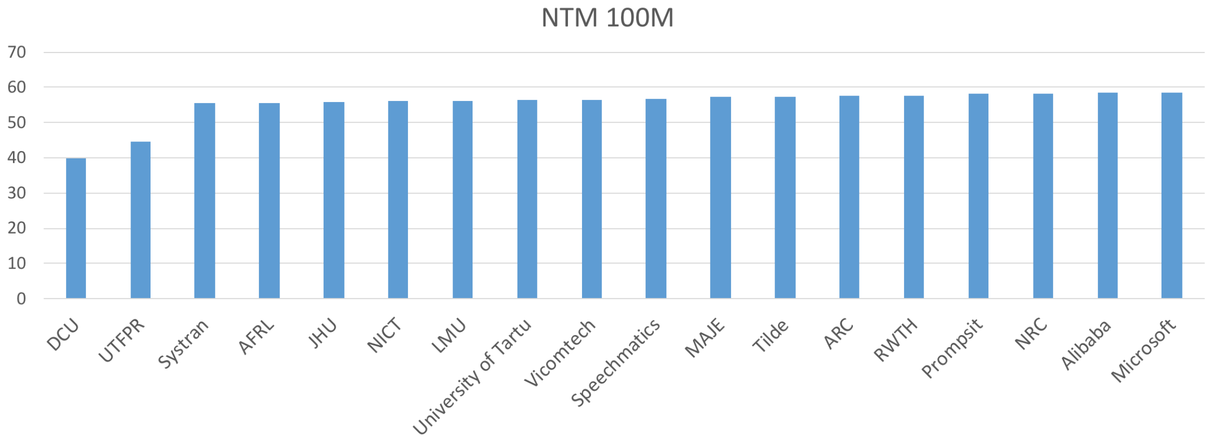


**Figure 2:** BLEU score of the best submission of each participant measured for the neural MT system trained with 100M tokens. Score is averaged over the six blind test sets.

the sentence pairs on the 1 billion word German-English Paracrawl corpus. Scores do not have to be meaningful, except that higher scores indicate better quality. The performance of the submissions is evaluated by sub-sampling 10 million and 100 million word corpora based on these scores, training statistical (Koehn et al., 2007) and neural (Junczys-Dowmunt et al., 2018) MT systems with these corpora, and assessing translation quality on six blind test sets[8] using the BLEU (Papineni et al., 2002) score.

Figure 2 displays the score of the best submission of each individual participant corresponding to the 100 million tokens corpus using the neural MT system. BLEU score is averaged over the six blind test sets.

As it can be seen, very similar results were obtained by most of the participants. Accuracy results fall within a margin of 3 points BLEU for the first 16 classified.

## 6 Conclusions

We have presented our submission to the WMT18 shared task on parallel corpus filtering. We participated for the first time using a neural sentence similarity classifier that predicts relatedness between sentence pairs in a multilingual context. The primary objective of our approach was to assess whether we were able to identify parallel sentences using a flexible method that relies on deep neural networks. Thus, eliminating the need for any domain specific feature engineering and using as few external tools as possible. We succeeded in our objective as we built a very simple network that was able to filter out divergent sentence pairs. Only assisted by a very simple filtering technique using rules based on length and language identification.

[8] Tests: newstest 2018, iwslt 2017, Acquis, EMEA, Global Voices, and KDE.

## References

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

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. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL 2007, System Demonstrations*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Joël Legrand, Michael Auli, and Ronan Collobert. 2016. Neural network-based word alignment through score aggregation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 66–73. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1310–III–1318. JMLR.org.

Minh Quang Pham, Josep Crego, Jean Senellart, and François Yvon. 2018. Fixing translation divergences in parallel corpora for neural mt. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*.

# Tilde's Parallel Corpus Filtering Methods for WMT 2018

**Mārcis Pinnis**
Tilde / Vienibas gatve 75A, Riga, Latvia
marcis.pinnis@tilde.lv

## Abstract

The paper describes parallel corpus filtering methods that allow reducing noise of noisy "parallel" corpora from a level where the corpora are not usable for neural machine translation training (i.e., the resulting systems fail to achieve reasonable translation quality; well below 10 BLEU points) up to a level where the trained systems show decent (over 20 BLEU points on a 10 million word dataset and up to 30 BLEU points on a 100 million word dataset). The paper also documents Tilde's submissions to the WMT 2018 shared task on parallel corpus filtering.

## 1 Introduction

Parallel data filtering for statistical machine translation (SMT) has shown to be a challenging task. Stricter filtering does not always yield positive results (Zariņa et al., 2015). This phenomenon can be explained with the higher robustness to noise of SMT systems, i.e., it does not harm the model if there are some incorrect translation candidates for a word or a phrase if the majority are still correct. However, there are also positive examples where data filtering allows improving SMT translation quality (Xu and Koehn, 2017). Neural machine translation (NMT), on the other hand, is much more sensitive to noise that is present in parallel data (Khayrallah and Koehn, 2018). From our own experience (as also shown by the experiments below), stricter filtering allows NMT models to show faster training tendencies and reach higher overall translation quality.

In this paper, we describe Tilde's methods for parallel data filtering for NMT system development and Tilde's submissions to the WMT 2018 shared task on parallel data filtering.

The paper is further structured as follows: Section 2 describes the data used in the filtering experiments, Section 3 provides details on the filtering methods that were applied to filter the parallel corpus of the shared task, Section 4 describes NMT experiments performed to evaluate the different filtering methods, Section 5 discusses the evaluation results, and Section 6 concludes the paper.

## 2 Data

The parallel data filtering experiments were performed on a German-English corpus that was provided by the WMT 2018 organisers. The corpus was a raw deduplicated subset[1] of the German-English ParaCrawl corpus[2]. It consists of one billion words and 104,002,521 sentence pairs.

For filtering, we require source-to-target and target-to-source probabilistic dictionaries. The dictionaries for the WMT 2018 experiments were acquired by 1) performing word alignment of the parallel corpora from the WMT 2018 shared task on news translation[3] (excluding the filtered ParaCrawl corpus) using *fast_align* (Dyer et al., 2013), and 2) performing raw probabilistic dictionary filtering using the transliteration-based probabilistic dictionary filtering method by Aker et al. (2014).

## 3 Filtering Methods

Although the filtering task required to score sentence pairs and not filter invalid sentence pairs out of the dataset, we start by filtering sentence pairs out of the raw corpus, after which we score each sentence pair and produce the scored output for submission. In order to filter the rather noisy "parallel" corpus, we use a combination of pre-existing parallel data filtering methods from the Tilde MT

[1] The corpus can be found online at http://www.statmt.org/wmt18/parallel-corpus-filtering.html.

[2] https://paracrawl.eu/download.html

[3] http://www.statmt.org/wmt18/translation-task.html

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 939–945
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64113

| | Filtering step | Sentence pairs | Proportion of the raw corpus |
|---|---|---|---|
| | *Raw corpus* | *104,002,521* | *100.00%* |
| ***Tilde MT filters for SMT systems*** | | | |
| 1.1. | Identical source and target sentence filter | 7,102,840 | 6.83% |
| 1.2. | Sentence length ratio filter | 5,276,660 | 5.07% |
| 1.3. | Maximum sentence length filter | 415,995 | 0.40% |
| 1.4. | Maximum word length filter | 286,485 | 0.28% |
| 1.5. | Maximum word count filter | 0 | 0.00% |
| 1.6. | Unique sentence pair filter | 20,821,646 | 20.02% |
| 1.7. | Foreign word filter | 14,983,927 | 14.41% |
| ***Additional Tilde MT filters for NMT systems*** | | | |
| 2.1. | Empty sentence filter | 222 | 0.00% |
| 2.2. | Token count ratio filter | 1,430,818 | 1.38% |
| 2.3. | Corrupt symbol filter | 33,519 | 0.03% |
| 2.4. | Digit mismatch filter | 20,534,497 | 19.74% |
| 2.5. | Invalid character filter | 630,818 | 0.61% |
| 2.6. | Invalid language filter | 1,229,434 | 1.18% |
| 2.7. | Stricter sentence length ratio filter | 1,710,401 | 1.64% |
| 2.8. | Low content overlap filter | 352,474 | 0.34% |
| ***Additional filters for the filtering task*** | | | |
| 3.1. | Non-translated sentence filter | 2,781,252 | 2.67% |
| 3.2. | Maximum alignment filter | 12,663,101 | 12.18% |
| | **Sentence pairs after filtering** | **13,748,432** | **13.22%** |

Table 1: Statistics of sentence pairs removed by individual filtering steps

platform (Pinnis et al., 2018) and methods specifically developed to address the noisy nature of the ParaCrawl corpus. Some of the filtering methods feature hyperparameters, which were set empirically in parallel corpora filtering experiments. The first part of the filters were originally developed to increase SMT system quality. The filters are applied in the following order (for statistics of each individual filtering step, refer to Table 1):

1. **Identical source and target sentence filter** - validates whether the source sentence and the target sentence in a sentence pair are not identical. Although it may very well be that a sentence translates into the same sentence, it is also a strong indicator of non-translated sentence pairs.

2. **Sentence length ratio filter**. The filter validates whether the longest sentence (in terms of characters) is less than three times longer than the shortest sentence. This filter is meant to identify partially translated sentences. However, it has to be noted that this filter has been tested only for language pairs with Latin-based, Cyrillic-based, and Greek alphabets.

3. **Maximum sentence length filter** - validates whether neither the source nor the target sentence is longer than 1000 characters long.

4. **Maximum word length filter** - validates whether neither the source nor the target sentence contains tokens that are longer than 50 characters and do not contain directory separator characters. When extracting data from, e.g., PDF or image files, it may happen that word boundaries are not captured correctly. This may result in long words being formed in sentences. This filter is intended to remove such sentence pairs.

5. **Maximum word count filter** - validates whether neither the source nor the target sentence contains more than 400 tokens.

6. **Unique sentence pair filter** - validates whether a sentence pair is unique. The shared task organisers claimed that deduplication was performed[4], however, this filter removes

[4] http://www.statmt.org/wmt18/parallel-corpus-filtering.html

all white-spaces and punctuation marks, replaces all digit sequences with a numeral placeholder, and lowercases the sentence before validating the uniqueness of a sentence pair. Therefore, it is able to identify more redundant data.

7. **Foreign word filter** - validates whether the source sentence contains only words written in the alphabet of the source language and whether the target sentence contains only words written in the alphabet of the target language.

The filtering steps, which had been originally developed for SMT systems, removed a total of 48,887,553 sentence pairs. After these steps, 55,114,968 sentence pairs were left in the corpus.

As NMT systems have shown to be more sensitive to noise (Khayrallah and Koehn, 2018), the Tilde MT platform implements additional filtering steps that are stricter compared to the previous filters. Together with the parallel data noise, these filters may also remove valid sentence pairs. However, as shown by the results in Section 5, the amount of the parallel data is less important than the quality of the data. The following are the additional filtering steps that are used when preparing data for NMT systems:

1. **Empty sentence filter** - validates whether neither the source nor the target sentence is empty (or contains only white-space characters) after decoding HTML entities.

2. **Token count ratio filter** - The filter validates whether the token count ratio of the shortest sentence and the longest sentence is greater than or equal to 0.3 (in other words, if one sentence has three times as many tokens as the other sentence, then the sentence pair is considered invalid).

3. **Corrupt symbol filter** - validates whether neither the source nor the target sentence contains words that contain question marks between letters (e.g., ‘*flie?en*’ instead of ‘*fließen*’, ‘*gr??ere*’ instead of ‘*größere*‘, etc.). Such words indicate encoding corruption in data, therefore, sentences containing such words are deleted.

4. **Digit mismatch filter** - validates whether all digits that can be found in the source sentence can also be found in the target sentence (and vice versa). Although this filter removes all sentence pairs where numbers that are written in digits have been translated into numbers written in words, it is effective for 1) identification of sentence breaking issues that are caused by incorrect handling of punctuation marks (e.g., cardinal numbers in some languages are written with the full stop character), and 2) identification of non-parallel content. By ensuring numeral writing consistency in parallel data, we can also ensure that digits will always be translated by the NMT systems as digits and numbers written in words as words.

5. **Invalid character filter** - validates whether neither the source nor the target sentence contains characters that have shown to indicate of encoding corruption issues. As most of potentially invalid (due to encoding corruption) sentence pairs are captured by the *foreign word filter* and the *corrupt symbol filter*, this filter provides just a minor addition - the list of invalid characters that are not included in valid alphabets consists of just four characters. However, this minor addition invalidates over 600 thousand sentence pairs.

6. **Invalid language filter** - validates whether the source sentence is written in the source language and whether the target sentence is written in the target language using a language detection tool (Shuyo, 2010). As language detection tools tend not to work well for shorter segments, this filter is applied only if the content overlap score (see below) between the source and target sentences is less than a trustworthy content alignment threshold (in the experiments set to 0.3) and the longest (source or target) sentence is at most two times longer than the shortest sentence.

7. **Stricter sentence length ratio filter** - validates whether the longest sentence (in terms of characters is less than two times longer than the shortest sentence.

8. **Low content overlap filter** - validates whether the content overlap according to the cross-lingual alignment tool *MPAligner* (Pinnis, 2013) is over a threshold. Because the content overlap metric produced by

*MPAligner* represents the level of parallelity, it is used to score sentence pairs. Therefore, the threshold was also set to a low value (0.01).

This far, a total of 74,809,736 were removed from the corpus, leaving a total of 29,192,785 sentence pairs remaining in the corpus.

When training NMT systems with the subsampled datasets, we identified that there were frequent (wrong) many-to-many alignments left in the corpus even after filtering. We also found that the corpus contained many entries with text in both languages on one side (i.e., imagine a translation where some of the source words are translated, but the majority is just copied over from the source segment and left untranslated), which contribute to parallel data noise. Therefore, we introduced two additional filters that address these issues:

1. **Non-translated sentence filter** - validates whether more than half of the source words have been translated (i.e., are not present in the target sentence).

2. **Maximum alignment filter** - keeps only those sentence pairs where the target sentence is the highest scored target sentence for the source sentence (according to the content overlap scores) and vice versa.

After all filtering steps, there were 13,748,432 sentence pairs left in the *Max Filtered+* corpus. In order to compare whether the full filtering workflow produces better results than a part of the workflow, we also prepared the following intermediate datasets:

1. *Filtered* - the corpus filtered up to and including the *low content overlap filter*. The dataset consists of 29,192,785 sentence pairs.

2. *Max Filtered* - the corpus filtered using all filters except the *Non-translated sentence filter*. The dataset consists of 15,613,062 sentence pairs.

3. *Filtered+* - the corpus filtered up to and including the *non-translated sentence filter*. The dataset consists of 26,411,533 sentence pairs.

4. *Max Filtered+ Rescored* - the corpus filtered using all filters and rescored by ranking sentences with a Round-robin-based method according to source sentence lengths. I.e., all sentence pairs were separated into different lists according to sentence lengths and sorted according to the content overlap scores in a descending order. Then, sentences were ranked by assigning the highest score to the best-scored unigram sentence, the second highest score to the best-scored bigram sentence, etc. We performed such rescoring, because the filtering assigned higher scores to shorter segments, thereby skewing the sentence length statistics towards shorter sentences. The dataset consists of 16,529,684 sentence pairs.

In each of the datasets (except for the *Max Filtered+ Rescored* dataset), sentence pairs were scored using the content overlap metric produced by *MPAligner*. In order to create scores for the raw dataset (i.e., to create submissions for the shared task), we scored each sentence pair in the raw dataset as follows: if a sentence pair was found in a particular filtered dataset, the sentence pair was scored using the score produced by *MPAligner* (or the rescoring method), otherwise the sentence pair received the score '*0*'. This means that all sentence pairs that were filtered out by any of the filtering steps, received the score '*0*'.

## 4 Trained Systems

To evaluate, which of the datasets allows achieving higher translation quality, we performed subsampling of the filtered datasets into 10 million and 100 million word datasets. For this, we used the *subselect.perl* script, which was provided by the organisers in the *dev-tools* package[5]. Then, we trained attention-based NMT systems with gated recurrent units in the recurrent layers using the Marian toolkit (Junczys-Dowmunt et al., 2018). All systems were trained using the configuration that is provided in the same package until convergence.

In addition to the filtered dataset systems, we trained four baseline systems. The first two baseline systems were trained on datasets, which were subsampled using the Hunalign (Varga et al., 2007) scores that were provided by the organisers. For the other two systems, data subsampling was performed on randomly assigned scores.

The NMT system training progress (in terms of BLEU scores on the raw tokenised development

[5] http://www.statmt.org/wmt18/parallel-corpus-filtering-data/dev-tools.tgz

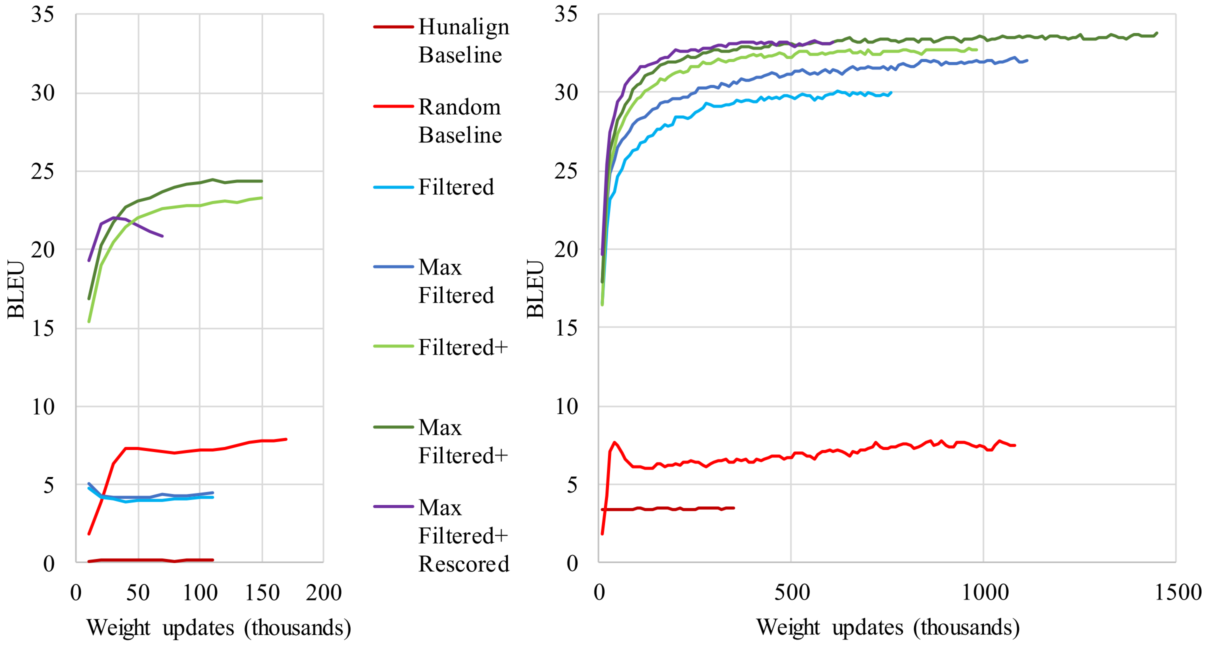


Figure 1: Training progress of NMT systems (10 million word systems - left; 100 million word systems - right)

set) is depicted in Figure 1. The figure shows that for the small dataset systems, only the systems with the *non-translated sentence filter* were able to achieve results of over 20 BLEU points. All other systems show rather poor performance, indicating the necessity of careful data cleaning. It is also evident that the *Filtered* and *Max Filtered* datasets contain too much noise among the highest scored sentence pairs. The reason for this is because the content overlap filter (by design) does not look at whether a sentence pair is a reciprocal translation. It tries to identify, just like a word alignment tool, which words in the source sentence correspond to which words in the target sentence, and non-translated words can be paired easily.

Although for the large dataset systems the *Filtered* and *Max Filtered* datasets contain higher levels of noise (compared to the more filtered datasets), they show comparative (however, lower) results to the more filtered datasets. The fact that the datasets are approximately 10 times larger than the smaller datasets allowed for higher quality sentence pairs to be included in the data sub-selected for NMT system training.

The figure also shows an interesting tendency for the *Max Filtered+ Rescored* dataset. In both experiments (10 million and 100 million word systems) the quality increases at the beginning, but then it starts to drop – very noticeably for the small system and slightly for the large system.

## 5 Results

Automatic evaluation results in terms of BLEU (Papineni et al., 2002) scores are provided in Table 2. For all systems, we used the '*test.sh*' script that was provided by the organisers in order to translate the test set and evaluate each model's translation quality.

The evaluation results illustrate the same dataset rankings as the training progress chart. The best results are achieved by using the *Max Filtered+* dataset.

We were also interested in seeing whether the filtering methods (by improving the parallel data quality) also allow improving out-of-vocabulary (OOV) word rates on the development set. It is evident in Table 2 that the OOV rate decreases by adding more filtering steps. However, there is one exception – the translation quality of the NMT systems, which were trained using the *Max Filtered+ Rescored* dataset, decreases although the OOV rate drops (especially when calculated for unique tokens). There may be multiple explanations for the quality decrease. For instance, for the smaller (10 million word) dataset, the rescoring introduced a higher percentage of lower quality sentence pairs due to the fact that the frequency of longer sentences is naturally lower than that of shorter sentences. E.g., there are 746,480 English

| System | BLEU | BLEU-C | Development data OOV rate (running) | (unique) |
|---|---|---|---|---|
| ***10 million token experiments*** | | | | |
| *Hunalign Baseline* | 0.15 | 0.14 | 8.27% | 32.08% |
| *Random Baseline* | 8.41 | 7.74 | 3.31% | 13.25% |
| Filtered | 4.86 | 4.32 | 6.25% | 25.28% |
| Max Filtered | 5.00 | 4.43 | 5.99% | 24.63% |
| Filtered+ | 21.35 | 19.75 | 4.54% | 18.44% |
| Max Filtered+ | **21.95** | **20.42** | 4.27% | 17.25% |
| Max Filtered+ Rescored | 20.10 | 18.75 | **3.29%** | **12.87%** |
| ***100 million token experiments*** | | | | |
| *Hunalign Baseline* | 3.64 | 3.28 | 1.78% | 7.16% |
| *Random Baseline* | 7.26 | 6.75 | 1.32% | 5.43% |
| Filtered | 27.72 | 26.14 | 1.39% | 5.65% |
| Max Filtered | 29.06 | 27.46 | 1.28% | 5.17% |
| Filtered+ | 30.24 | 28.59 | 1.32% | 5.24% |
| Max Filtered+ | **30.83** | **29.14** | **1.31%** | 5.10% |
| Max Filtered+ Rescored | 30.40 | 28.78 | 1.32% | **4.95%** |

Table 2: Evaluation results of NMT systems trained using different sub-sampled filtered datasets (the table shows case-insensitive BLEU and case-sensitive BLEU (BLEU-C))

sentences that consist of five tokens, compared to just 2673 sentences of 80 tokens in the *Max Filtered+* dataset (which was used to acquire the rescored dataset). This means that the rescoring method was forced to select lower quality longer sentence pairs simply because of insufficient sentence pairs to select from. For the larger dataset, the results also show that the running OOV rate is slightly larger than the unique token OOV rate. However, the issue with the limited number of longer sentences did affect also the larger system as the sub-sampled dataset included all sentence pairs that were longer than or equal to 42 tokens regardless of their quality. For future work, it could be beneficial to investigate whether a fixed content overlap threshold could allow the rescoring method to perform better.

For the WMT 2018 shared task, we submitted the following three datasets:

1. *tilde-isolated (Filtered+)* – this dataset represents isolated sentence filtering where only individual sentence pairs are passed to the filtering method.

2. *tilde-max (Max Filtered+)* – this dataset represents full corpus filtering where (in addition to the filtering results of a particular sentence pair) also information about other sentence pairs is used to decide whether to keep a sentence pair or not.

3. *tilde-max-rescored (Max Filtered+ Rescored)* – this dataset represents both full corpus filtering and (a rather simple) data selection method.

## 6 Conclusion

The paper presented parallel corpus filtering methods that allow reducing the noise in noisy "parallel" corpora to a level where the corpus is usable in neural machine translation system development. Most of the filtering methods are simple (except for the *low content overlap filter*) and do not require any machine learning methods to be implemented (except for the *invalid language filter*). We showed that, by applying stricter filtering methods, NMT system quality increases.

For the WMT 2018 shared task on corpus filtering, we submitted three scored datasets that represent isolated sentence filtering (*Filtered+*), full corpus filtering (*Max Filtered+*), and (a rather simple method for) full corpus filtering with data selection (*Max Filtered+ Rescored*).

The filtering methods are integrated into the Tilde MT platform and serve its users when they require SMT and NMT system training.

For future work, it may be beneficial to perform ablation experiments, to identify, which of the individual filtering methods contributes the most in order to acquire a higher quality parallel corpus.

## Acknowledgments

The research has been supported by the European Regional Development Fund within the research project "Neural Network Modelling for Inflected Natural Languages" No. 1.1.1.1/16/A/215.

## References

Ahmet Aker, Monica Lestari Paramita, Mārcis Pinnis, and Robert Gaizauskas. 2014. Bilingual Dictionaries for All EU Languages. In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC'14)*, pages 2839–2845, Reykjavik, Iceland. European Language Resources Association (ELRA).

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013)*, June, pages 644–648, Atlanta, USA.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Mārcis Pinnis. 2013. Context Independent Term Mapper for European Languages. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2013)*, pages 562–570, Hissar, Bulgaria.

Mārcis Pinnis, Andrejs Vasiļjevs, Rihards Kalniņš, Roberts Rozis, Raivis Skadiņš, and Valters Šics. 2018. Tilde MT Platform for Developing Client Specific MT Solutions. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Nakatani Shuyo. 2010. Language detection library for java.

Daniel Varga, Peter Halacsy, Andras Kornai, Viktor Nagy, Laszlo Nemeth, and Viktor Tron. 2007. Parallel corpora for medium density languages. *Recent Advances in Natural Language Processing IV: Selected Papers from RANLP 2005*, 292:247.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950.

Ieva Zariņa, Pēteris Ņikiforovs, and Raivis Skadiņš. 2015. Word Alignment Based Parallel Corpora Evaluation and Cleaning Using Machine Learning Techniques. In *Proceedings of the Eighteenth Annual Conference of the European Association for Machine Translation (EAMT 2015)*, pages 185–192, Antalya. European Association for Machine Translation.

# The RWTH Aachen University Filtering System for the WMT 2018 Parallel Corpus Filtering Task

**Nick Rossenbach, Jan Rosendahl, Yunsu Kim,**
**Miguel Graça, Aman Gokrani, Hermann Ney**
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
<surname>@i6.informatik.rwth-aachen.de

## Abstract

This paper describes the submission of RWTH Aachen University for the De→En parallel corpus filtering task of the *EMNLP 2018 Third Conference on Machine Translation* (WMT 2018). We use several rule-based, heuristic methods to preselect sentence pairs. These sentence pairs are scored with count-based and neural systems as language and translation models. In addition to single sentence-pair scoring, we further implement a simple redundancy removing heuristic. Our best performing corpus filtering system relies on recurrent neural language models and translation models based on the transformer architecture. A model trained on 10M randomly sampled tokens reaches a performance of 9.2% BLEU on newstest2018. Using our filtering and ranking techniques we achieve 34.8% BLEU.

## 1 Introduction

In this work we describe the corpus filtering system of the RWTH Aachen University for the WMT 2018 parallel corpus filtering task.

We decided to rank the data using a two-stage process. During the first stage, we reduce the number of parallel sentences by applying basic rule-based heuristics each of whom can reject a sentence as described in Section 3. Afterward, we apply a variety of models on the remaining sentences to assign a score to each sentence pair. The details of those models, namely language models and translation models, can be found in Section 4.

Our final submission consists of three different systems on top of rule-based filtering: Two of them are based on scoring each sentence pair independently using either only count-based models or only neural models. The third submission extends on the neural network-based submission by removing redundancies before ranking the sentences.

We compare the behavior of neural network based models to count-based models and find that the performance differs by more than 1.0 % BLEU on average across all test sets. In total our best system reaches a performance of 34.8 % BLEU compared to 9.2 % BLEU using random sampling on newstest2018 of the news translation task with 10M token subsampled training data. We report our findings and results in detail in Section 6.

## 2 Preprocessing

As a first step, we normalize the data by removing soft-hyphen and zero-width space symbols. Furthermore, we replace all hash symbols (#) because we use them as separation symbol. A language specific tokenizer from Moses (Koehn et al., 2007) is applied to both sides of the corpus. We later found out that this language specific splitting can cause some issues if equal patterns are not split the same way on source and target side (see Section 3.6).

After tokenization, we search for and replace any escaped characters with the corresponding symbol and squeeze repeating whitespaces. We true-case our words by applying a frequent casing model from the Jane toolkit (Vilar et al., 2010) based on the parallel corpora.

If data is used to train the count-based models or if we apply the count-based models on a sentence pair, numbers are replaced by a category symbol. For the neural models, we generate joint BPE merge operations on the parallel training data with 20k merge operations (Sennrich et al., 2016). For the training of the neural scoring models, we create BPE vocabularies based on the clean parallel data of the WMT news task[1], and use a vocabulary threshold of 50. For evaluation of the subsampled data, we did not use a vocabulary threshold.

[1] CommonCrawl, Europarl, NewsCommentary, Rapid

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 946–954
Belgium, Brussels, October 31 - Novermber 1, 2018.
https://doi.org/10.18653/v1/W18-64114

## 3 Rule-based Filtering

As the scoring of 104 million sentence pairs is hardly feasible with computationally expensive models like a transformer model (Vaswani et al., 2017), we need to preselect a smaller subset of the data. We do this by applying several rule-based heuristics as a first stage of our data cleaning pipeline. A sentence pair is removed from the corpus if its source side or target side fail to obey any of these rules. Note that none of these rules is language specific to either German or English and that they place only very mild assumptions on what 'good data' should look like.

Besides reducing the amount of data, some of the heuristic filtering methods can deal with aspects that can not be captured with language models and are hard to cover by translation models (see Subsection 6.4).

Table 1 shows the amount of remaining sentence pairs and tokens after applying each heuristic in sequential order.

### 3.1 Minimum Words

Our first heuristic filter ensures that every sentence contains at least a certain number of words. To do so, we count the number of tokens (i.e. character sequence between two spaces) that contain at least one letter from the alphabet of the language. Thus numbers or punctuation symbols are not counted as words according to this definition. A sentence is only valid if this number reaches a certain threshold (which we set to 3 for all our experiments).

### 3.2 Average Word Length

In the next step of the filtering process, we remove long chains of characters and sequences where only single characters appear. This aims in particular for lines which consist mainly of a single, very long URL. Although we did not expect a lot of sentence pairs to have an average word length lower than 2 or bigger than 20 characters, we removed about 1% of the sentence pairs with this procedure.

### 3.3 Length Ratio

Judging sentence pairs by the ratio of source sentence vs. target sentence length is a very simple but effective criterion. We limited this length ratio to be not greater than 1.7. Because of tokenization, all punctuation symbols are counted as single words. To smooth the ratio for shorter sentences, we always add 1 to the token count, i.e. we reject the sentence if:

$$\frac{J+1}{I+1} > 1.7 \vee \frac{I+1}{J+1} > 1.7$$

where $I$ is the target and $J$ the source sequence length.

### 3.4 Maximum Sentence Length

Because many translation systems have an upper bound for the sentence length during training and to reduce the computational cost of our scoring models, we limited the maximum number of tokens to 50.

### 3.5 Maximum Subword-Token Length

As scoring with Sockeye (Hieber et al., 2017) transformer model requires a maximum sequence length as fixed parameter, we enforce a limit on the number of subword units. The subword merge operations are computed on the parallel WMT 2018 news training data, excluding the filtered ParaCrawl data. We limited each sentence to consist of a maximum of 100 subword tokens.

### 3.6 Levenshtein Distance

In our experiments we observe that the transformer model tends to assign a very high score to sentence pairs in which source and target share a great number of words. This happens even if neither the given source nor the given target sentence are in the correct language. It seems that the model regards copying as a valid form of translation. To detect sentences where source and target are too similar, we compute the word-level Levenshtein distance $D$ (Levenshtein, 1966) between the lowercased sentences . We also take into account a length normalized Levenshtein distance $\bar{D} = \frac{D}{I+J}$. A sentence is rejected if:

$$D \leq 1 \vee \bar{D} \leq 0.15$$

These values were determined by visually looking at 100k random examples ranked by $\bar{D}$ and ensuring that no valid looking sentence gets removed. The language specific tokenizers sometimes split the same sequence differently depending on the language, which increases the distance e.g.:

*the do ' s and don ' ts of the audience .*
*the do 's and don 'ts of the audience .*

| Method | pairs | de tok. | en tok. | del.% | del. total |
|---|---|---|---|---|---|
| Original Data | 104.0M | 1,520M | 1,562M | | |
| Min. Words (3.1) | 61.9M | 1,276M | 1,313M | 40.45% | 42.0M |
| Avg. Word Length (3.2) | 61.3M | 1,262M | 1,298M | 0.94% | 0.6M |
| Length Ratio (3.3) | 50.6M | 1,072M | 1,092M | 17.60% | 10.8M |
| Max. Seq. Length (3.4) | 46.0M | 625M | 642M | 8.91% | 4.5M |
| Max. Seq. Length (BPE) (3.5) | 46.0M | 622M | 638M | 0.20% | 0.1M |
| Levenshtein (3.6) | 36.6M | 512M | 528M | 20.26% | 9.3M |
| Word Token Ratio (3.7) | 28.1M | 398M | 412M | 23.38% | 8.6M |
| Redundancy (3.8) | 13.0M | 227M | 236M | 53.84% | 15.1M |

Table 1: Sizes of datasets after applying the heuristic filtering methods. Sizes are given in sentence pairs, tokens on German side and tokens on English side. Every heuristic is applied on top of the preceding heuristic. The last two columns show the percentage (with respect to its input not the original corpus) respectively the absolute number of lines removed by a heuristic.

Thus, the Levenshtein heuristic sometimes misses some sentence pairs that should have been removed.

### 3.7 Word Token Ratio

We extend the idea of minimum word filtering from Section 3.1 to scale with sentence length. We count the number of tokens that contain at least one character that is a standard alphabet letter. If this count is less than 60% of the total sentence length, we reject the sentence. This can be helpful to remove sentences from languages with different alphabets or lines which simply consist of a time and date. Also, sentences with more than 60% numbers and punctuation symbols are removed.

### 3.8 Redundancy

To increase the amount of information in the subsampled data, we wanted to remove redundant information. Checking the redundancy of a sentence in the context of a big corpus is challenging, as trivial algorithms need to do $\frac{C^2}{2}$ comparisons for corpus size $C$ which is not feasible for large datasets. One simple solution for removing identical sentences in linear time is to compute a hash value[2] for each sentence, and check for existing hashes in a set. We extended this approach to detect 'similar' sentences by removing each word individually and store the hash of the remaining sentence. By doing this we also remove sentences that have a word edit distance of one compared to any previously added sentence. We do not distinguish between source or target side sentences, both are stored in the same set. A simple pseudocode description is shown in Algorithm 1.

**Algorithm 1:** Duplicate checking

```
$hm \leftarrow empty\_hashmap()$
for each sentence $s_1^N$ do
    $sent\_hm \leftarrow empty\_hashmap()$
    for each position $i \in [1, N]$ do
        $h \leftarrow hash([s_1^{i-1}, s_{i+1}^N])$
        if $h \in hm$ then
            reject $s_1^N$
            break
        else
            $sent\_hm$.add($h$)
    if $s_1^N$ not $rejected$ then
        $hm$.add($sent\_hm$)
```

## 4 Model-based Scoring

In the second stage of our filtering pipeline we score each sentence using different kinds of language and translation models. Every model assigns a probability to each sentence. These scores are used afterward to rank the corpus and select the top sentences.

### 4.1 Count-Based Language Model

To score the remaining sentences, we start by applying count based language models on each side of the parallel sentences. The language models used are 5-gram KenLM (Heafield et al., 2013) models with singleton tri-gram pruning and trained with modified interpolated kneser-ney smoothing (Chen and Goodman, 1996). They

[2] We use the python3 default hash() function

are trained on the NewsCrawl 2016, Europarl, NewsCommentary and Rapid corpora from the WMT 2018 German→ English task. Adding NewsCrawl 2012-2015 as further monolingual training sets does not achieve better results.

We apply the preprocessing mentioned in Section 2 and we remove any sentence from the training data that contains token repetitions of length three or more. This is done to get rid of phenomenons like chains of exclamation marks. For more details about the data selection see Section 6.2.

### 4.2 IBM1 Dictionary Model

IBM1 models are a simple approach to model the dependency $p(e_1^I|f_1^J)$, as they assume a uniform alignment. We train the model with the GIZA++ toolkit (Och and Ney, 2003) on the parallel data to create an IBM1 table. IBM model 1 scores are computed as in (Brown et al., 1993):

$$p(e_1^I|f_1^J) = \frac{1}{(J+1)^I} \prod_{i=1}^{I} \sum_{j=0}^{J} p(e_i|f_j) \quad (1)$$

where $I$ and $J$ are the length of the target respectively source sentence, and $f_0$ is a null token. We train IBM1 models for both directions (s2t and t2s) using the bilingual data from the WMT 2018 German↔English task namely the Europarl, CommonCrawl, NewsCommentary and Rapid corpus.

### 4.3 Neural Network Language Model

We modified the RWTH Aachen translation system as described in (Peter et al., 2017) based on the Blocks framework (van Merriënboer et al., 2015) and Theano (Theano Development Team, 2016) to also work as a recurrent language model. The training data is chosen to be equivalent to the one used in the training of the count-based models. The language model has an embedding size of 250 and two LSTM layers (Hochreiter and Schmidhuber, 1997) with a hidden size of 1000. As it is default in Blocks, it also includes a maxout layer of factor 2 (Goodfellow et al., 2013) between the second LSTM and the output softmax. The system was trained using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.001 for 300k iterations with a batch size of 100 sentences and a dropout (Srivastava et al., 2014) of 0.2.

### 4.4 Transformer Translation Model

As neural network-based translation model we use the transformer architecture (Vaswani et al., 2017) implemented in the Sockeye toolkit (Hieber et al., 2017) which is build on top of MXNet (Chen et al., 2015). Encoder and decoder each consist of 6 layers. The hidden and embedding size is set to be 512 and the feed forward layer size is 2048. The number of attention heads is 8. A dropout of 0.1 is applied, except for the embedding layer. We use an initial learning rate of 0.0002. We save checkpoints every 20k iterations, and reduce the learning rate by factor 0.7 after each non-improving checkpoint (measured by means of perplexity on newstest2015). The network is trained on the bilingual data from the WMT 2018 German↔English task namely the Europarl, CommonCrawl, NewsCommentary and Rapid corpus.

## 5 Evaluation Model

To check the quality of a filtering approach, we train a transformer model on the top 10M respectively top 100M subwords of the scored training data. We mainly focus on the 10M-subsampling results, as this scenario shows clearer differences in performance between different methods. Like in Section 4.4 we use the Sockeye implementation of the transformer architecture but we train smaller models for evaluation purposes. The decision to use small transformer networks was made as they give strong results in a much shorter amount of time (1 day compared to 5 days). To verify the generality of the approach we cross-checked several experiments using recurrent neural network-based (RNN-based) translation systems from the Marian framework (Junczys-Dowmunt et al., 2018) and found their training behavior to be correlated.

We apply the provided subsampling script on the filtered data to extract training data. Due to an error in our filtering setup the input data is tokenized and subworded. Since both procedures increase the number of tokens per sentence, we extract less sentence pairs than intended.

Note that because of these two effects (transformer and lower subsampling rate) BLEU scores reported in this submission can vary in comparison to other submissions, if RNN-based systems are used.

For the transformer model used for evaluation, we use only 3 layers in encoder and decoder and increase the batch size to 8,000 words. We train for 100k updates evaluating a checkpoint every 10k updates for the 10M word experiments. In the case of 100M words of training data, 200k update steps are performed with a checkpoint being written every 20k updates. The best checkpoint is selected by computing BLEU (Papineni et al., 2002) on newstest2015. The beam-size for translation is 12. We report BLEU scores using mteval from the Moses toolkit (Koehn et al., 2007) and TER scores (Snover et al., 2006) using TERcom.

During system building most of our design decisions are based on results for the 10M-word-version of the task, however, we observe very similar trends for the 100M-word subsampling. For brevity we report most results only on the smaller subsamples.

## 6 Experimental Evaluation

In this section we report the results of our filtering experiments. We use newstest2015 and newstest2017 as development sets and report the results on newstest2018. For brevity, we shorten the names of newstestX to tstX in the header of several tables.

All BLEU and TER scores reported in this sections are obtained by using the system under consideration as filtering system and training the transformer system described in Section 5 on the resulting training data. All processing steps and experiments are organized with Sisyphus (Peter et al., 2018) as workflow manager.

### 6.1 Rule-based Filtering

The purpose of the rule-based heuristics is not to select perfect training data, but rather to reduce the original 104M lines of the ParaCrawl corpus down to an amount that can be handled by stronger, computationally more complex, methods. Table 2 and Table 3 show the evaluation results for different levels of heuristic cleaning for 10M subsampling and 100M subsampling respectively. Since there is no score-based ranking yet, we sample the desired amount of data randomly from the filtered corpus. Although a big part of the corpus is removed (58M sentences or 60% of the original corpus), the first 5 heuristic steps (3.1)-(3.5) have nearly no impact on the data quality. Applying the Levenshtein distance heuristic (see Section 3.6) resulted in a strong increase of data quality to an average of 20.2% BLEU. This increase occures despite removing only 20% of the sentence pairs. Applying the word-token-ratio-heuristic (see Section 3.7) has a lesser impact, but still increases the evaluation scores by about 1.0% BLEU for 10M and about 0.6% BLEU for 100M subsampled data. Checking for redundant sentences increases the scores by up to 3.9% BLEU. This is not suprising, because as more than 50% of the sentences are removed, we replace 50% of the random selected data by potentially more informative examples.

| Filtering | tst15 | | tst17 | | tst18 | |
|---|---|---|---|---|---|---|
| 10M | BLEU | TER | BLEU | TER | BLEU | TER |
| Unfiltered | 8.3 | 87 | 8.3 | 87.3 | 9.2 | 85 |
| (3.1) - (3.5) | 8.4 | 82.1 | 8.7 | 82.1 | 10.2 | 79.4 |
| (3.1) - (3.6) | 18.7 | 64.4 | 19.1 | 65.1 | 22.8 | 59.0 |
| (3.1) - (3.7) | 20.1 | 61.5 | 20.0 | 62.7 | 25.0 | 56.0 |
| (3.1) - (3.8) | 23.3 | 56.7 | 23.5 | 57.3 | 28.9 | 50.6 |

Table 2: Model evaluation of 10M random sampling from the datasets created by rule-based heuristic filtering.

| Filtering | tst15 | | tst17 | | tst18 | |
|---|---|---|---|---|---|---|
| 100M | BLEU | TER | BLEU | TER | BLEU | TER |
| Unfiltered | 9.1 | 84.7 | 8.6 | 85.2 | 10.6 | 80.9 |
| (3.1) - (3.5) | 10.8 | 78.2 | 10.3 | 79.5 | 12.6 | 75.5 |
| (3.1) - (3.6) | 23.2 | 59.0 | 23.2 | 60.8 | 29.1 | 52.4 |
| (3.1) - (3.7) | 23.8 | 57.9 | 23.8 | 59.4 | 30.0 | 50.8 |
| (3.1) - (3.8) | 27.2 | 53.1 | 27.3 | 53.5 | 33.8 | 45.8 |

Table 3: Model evaluation of 100M random sampling from the datasets created by rule-based heuristic filtering.

### 6.2 Model-based Scoring

While the heuristics alone already result in quite satisfying cleaning results, the scoring models are used to create a ranking of the remaining sentences.

We use the corpus cleaned by the heuristics (3.1)-(3.6) as starting point for the following experiments.

In our first experiments we test the behavior of the models presented in Section 4 in isolation. Note that all our language model experiments always rely on a source and a target side language model each scoring the corresponding part of the sentence pair. All experiments with IBM1 or transformer models use a combination of a source-to-target and a target-to-source model. We average the log probabilities of the models to get a single

| | System (10M) | **newstest15** | | **newstest17** | | **newstest18** | |
|---|---|---|---|---|---|---|---|
| | | BLEU | TER | BLEU | TER | BLEU | TER |
| #1 | (3.1)-(3.6) random sampling | 18.7 | 64.4 | 19.1 | 65.1 | 22.8 | 59.0 |
| #2 | KenLM | 21.3 | 62.1 | 21.2 | 63.6 | 25.8 | 56.5 |
| #3 | BlocksLM | 23.3 | 59.6 | 23.2 | 60.9 | 28.1 | 54.6 |
| #4 | IBM | 24.7 | 55.2 | 25.2 | 55.3 | 31.3 | 47.7 |
| #5 | Transformer | 24.2 | 55.8 | 24.2 | 56.3 | 30.2 | 48.7 |
| #6 | KenLM + IBM | 26.8 | 53.8 | 26.9 | 54.3 | 33.0 | 46.7 |
| #7 | + Word Token Ratio (3.7) | 26.6 | 53.9 | 27.0 | 54.0 | 33.1 | 46.2 |
| #8 | + Redundancy (3.8) | 27.2 | 53.5 | 27.1 | 53.9 | 33.4 | 46.2 |
| #9 | + **IBM retraining**[1] | 27.2 | 53.3 | 27.6 | 53.4 | 33.4 | 46.1 |
| #10 | BlocksLM + IBM | 27.2 | 53.6 | 27.4 | 53.7 | 33.5 | 46.1 |
| #11 | BlocksLM + Transformer | 28.1 | 52.4 | 28.4 | 52.4 | 34.6 | 45.0 |
| #12 | + **Word Token Ratio (3.7)**[2] | 28.0 | 52.6 | 28.3 | 52.6 | 34.4 | 45.1 |
| #13 | + **Redundancy (3.8)**[3] | 28.1 | 52.3 | 28.3 | 52.3 | 34.8 | 44.8 |
| #14 | KenLM + IBM + BlocksLM + Trans. | 27.5 | 53.0 | 27.8 | 53.7 | 33.5 | 46.0 |

Table 4: Results for 10M word subsampling when applying different scoring models on already filtered data. All models are scoring the data that was filtered with methods described in Section 3.1 to 3.6. For model-based filtering, both source and target sides are scored.
[1]: Submission 1 with name **rwth-count**
[2]: Submission 2 with name **rwth-nn**
[3]: Submission 3 with name **rwth-nn-redundant**

score, where 0 is the best and all other scores are negative. For our submission, we added a score of -1000 for rejected sentence pairs.

From Table 4 we can see that all 4 trained models improve the heuristic filtering by more than 2.0% BLEU. Note that BlocksLM achieves better filtering results than the count-based KenLM system. However neural systems provide weaker cleaning when it comes to translation models. We are not sure why transformer performs up to 1.1% BLEU and 1.0% TER worse than IBM1 models in standalone comparison. A possible explanation is that the transformer model prefers very short sentences when not combined with a language model. For 10M subsampling, the IBM1 model ranks sentences with an average sentence length of 20 as best, while for the transformer model it is only 10.6. Combined with a langauage model, this value increases to 17.6. As can be seen from Table 4 Row #10 vs #11 this effect disappears when both systems are extended with the same language model. In this case the purely neural-network-based system has a consistent lead of roughly 1.0 % BLEU.

From Table 4 we observe that language models generally perform worse in cleaning than translation models. This could be due to the fact that many kinds of noise, which can be detected by only looking at either the source or the target sentence, are already removed by the heuristics.

Combining KenLM with an IBM1 model improves the BLEU score by 1.8% on average over IBM1 models and by 6.1% BLEU over KenLM. Adding the word to token ratio (3.7) does not affect the system performance. Note that word to token ratio was quite effective when only heuristic filtering is used (Table 2). This underlines the assumption that our heuristics remove sentence pairs, which would be sorted out by trained models anyhow. To close the gap between the count-based and neural-network-based filtering, we retrain the IBM model using its original training data plus the top 500k sentences selected from the to-be-cleaned ParaCrawl corpus, which was filtered using transformer. This improves the system by up to 0.6% BLEU but the results are still more than 0.7% BLEU behind a similar neural-network based filtering system (see Table 4 Row #9 vs. Row #12).

We achieve the best performance by combining the BlocksLM with the transformer translation systems plus the word token ratio and redundancy

| System | perplexity | | tst17 eval | |
|---|---|---|---|---|
| | de | en | BLEU | TER |
| KenLM | 282.2 | 150.5 | 26.9* | 54.3* |
| + CommonCrawl | 277.6 | 146.6 | 25.8* | 55.6* |
| BlocksLM | 111.07 | 120.62 | 27.4* | 53.7* |
| + CommonCrawl | 110.55 | 117.7 | 26.4* | 54.5* |

Table 5: Comparison of language model perplexity with its performance as data cleaning system as well as the effect of CommonCrawl on LMs.
* For KenLM filtering results we combine the corresponding LM scores of source, target and two fixed IBM1 scores.

heuristic. The resulting system uses heuristics to filter a corpus of 104M lines down to 13M sentence pairs without the need to apply any complex model. This part of the pipeline is cheap and fast, and already gives a performance of 23.3 % BLEU on newstest2015 (see Table 2). Applying strong translation and language models yields an additional improvement of 4.8% BLEU as is shown in Table 4.

### 6.3 Noisy Data Effect

To investigate the effect of noisy training data for the scoring models, we add the CommonCrawl corpus to the language model training data. Although the perplexity on the dev set improves slightly for both model architectures (see Table 5), the evaluation results for the subsampled data drop by about 1.0 % BLEU. This indicates that the models are required to not only recognize good sentences well, but also to give low scores to bad sentences. If the training data contains more noisy data, a model will give higher scores to bad sentences. While this is usually a smaller problem for translation models, in terms of sentence ranking it is an important issue.

### 6.4 Levenshtein Distance

Table 6 shows the effect of the Levenshtein heuristic on count-based and neural scoring models. While removing sentence pairs with similar source and target does not change the performance when ranking with count-based models, it increases the performance of neural models by up to 1.0% BLEU. This confirms the assumption from Section 3.6 that transformer-based models assign high scores when copying sentences. We regard Levenshtein-based filtering as a crucial heuristic when ranking sentence pairs with neural models.

### 6.5 Submission Results

Table 7 shows the official evaluation results of our submitted rankings compared to the best submission from Microsoft. While slightly exceeding on the SMT 10M evaluation, we are 0.8% BLEU behind the leading submission on NMT 100M. For NMT 10M, we have the best results on newstest2018, iwslt2017 and Acquis, but perform a lot weaker on KDE, thus being worse on average. This might be due to some unavoidable domain adaptation when training language models with mono-lingual news data.

## 7 Conclusion

This paper describes the RWTH Aachen University data-filtering and ranking methods for the WMT 2018 parallel corpus filtering task. We describe various rule-based heuristic filtering methods to reduce the amount of data to be scored, and to tackle some of the weak spots of neural language and translation models. We describe 4 different ranking models, two language model architectures and 2 translation models, count-based and neural. Our results indicate that even without ranking the sentence pairs with model scores, a high quality subset can be extracted.

Among the submissions our best models works very well for the small data condition, ranking first on the 10M-subsampled SMT translation and second on the 10M-subsampled NMT translation. Also with the 100M-subsampled data condition, we perform above average, with a gap of 0.7% average BLEU to the leading submission for NMT translation.

## Acknowledgements

This work has received funding from the European Research Council (ERC) (under the European Union's Horizon 2020 research and innovation programme, grant agreement No 694537, project ”SEQCLAS”) and the Deutsche Forschungsgemeinschaft (DFG; grant agreement NE 572/8-1, project ”CoreTec”). The GPU computing cluster was supported by DFG (Deutsche Forschungsgemeinschaft) under grant INST 222/1168-1 FUGG.
The work reflects only the authors' views and

| System (10M) | **newstest15** | | **newstest17** | | **newstest18** | |
|---|---|---|---|---|---|---|
| | BLEU | TER | BLEU | TER | BLEU | TER |
| (3.1-5) + KenLM + IBM | 26.8 | 53.9 | 27.0 | 54.3 | 32.5 | 46.9 |
| (3.1-5) + KenLM + IBM + Lev.Sht. | 26.8 | 53.8 | 26.9 | 54.3 | 33.0 | 46.7 |
| (3.1-5) + BlocksLM + Transformer | 27.3 | 53.2 | 27.4 | 53.5 | 33.6 | 46.1 |
| (3.1-5) + BlocksLM + Transformer + Lev.Sht. | 28.1 | 52.4 | 28.4 | 52.4 | 34.6 | 45.0 |

Table 6: Effect of using the Levenshtein distance heuristic (3.6) on count-based and neural scoring.

| Submission System | SMT 10M | SMT 100M | NMT 10M | NMT 100M |
|---|---|---|---|---|
| (3.1)-(3.7) + KenLM + retrained IBM1[1] | 23.85 | 25.91 | 26.65 | 31.05 |
| (3.1)-(3.7) + BlocksLM + Transfomer[2] | 24.53 | 26.18 | 28.00 | 31.20 |
| + Redundancy Heuristic[3] | 24.58 | 26.21 | 28.01 | 31.29 |
| Microsoft | 24.45 | 26.50 | 28.62 | 32.06 |

Table 7: Official submission result for each evaluation method. The scores report the average BLEU % across all 6 test sets.
[1]: Submission 1 with name **rwth-count**
[2]: Submission 2 with name **rwth-nn**
[3]: Submission 3 with name **rwth-nn-redundant**

none of the funding agencies is responsible for any use that may be made of the information it contains.

## References

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*. Version 1.

Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. 2013. Maxout networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1319–III–1327. JMLR.org.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1712.05690*. Version 2.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. Version 9.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantine, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *acl07-poster*, pages 177–180, Prague, Czech Republic.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

Bart van Merriënboer, Dzmitry Bahdanau, Vincent Dumoulin, Dmitriy Serdyuk, David Warde-Farley, Jan Chorowski, and Yoshua Bengio. 2015. Blocks and Fuel: Frameworks for deep learning. *CoRR*, abs/1506.00619.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Jan-Thorsten Peter, Eugen Beck, and Hermann Ney. 2018. Sisyphus, a workflow manager designed for machine translation and automatic speech recognition. In *2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. (To appear).

Jan-Thorsten Peter, Andreas Guta, Tamer Alkhouli, Parnia Bahar, Jan Rosendahl, Nick Rossenbach, Miguel Graça, and Ney Hermann. 2017. The RWTH Aachen University English-German and German-English machine translation system for WMT 2017. In *EMNLP 2017 Second Conference on Machine Translation*, Copenhagen, Denmark.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. 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*, pages 223–231, Cambridge, Massachusetts, USA.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688. Version 1.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open source hierarchical translation, extended with reordering and lexicon models. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 262–270. Association for Computational Linguistics.

# Prompsit's submission to WMT 2018 Parallel Corpus Filtering shared task

**Víctor M. Sánchez-Cartagena, Marta Bañón,**
**Sergio Ortiz-Rojas, Gema Ramírez-Sánchez**
Prompsit Language Engineering
Av. Universitat s/n. Edifici Quorum III
E-03202 Elx, Spain
{vmsanchez,sortiz,gramirez,mbanon}@prompsit.com

## Abstract

This paper describes Prompsit Language Engineering's submissions to the WMT 2018 parallel corpus filtering shared task. Our four submissions were based on an automatic classifier for identifying pairs of sentences that are mutual translations. A set of hand-crafted hard rules for discarding sentences with evident flaws were applied before the classifier. We explored different strategies for achieving a training corpus with diverse vocabulary and fluent sentences: language model scoring, an active-learning-inspired data selection algorithm and $n$-gram saturation. Our submissions were very competitive in comparison with other participants on the 100 million word training corpus.

## 1 Introduction

This paper describes the systems submitted by Prompsit Language Engineering[1] to the parallel corpus filtering shared task (Koehn et al., 2018) featured in the Third Conference on Machine Translation (WMT 2018).

Given a very noisy 1 billion-word German-English parallel corpus crawled from the web,[2] participants have to subselect sentence pairs that amount to (a) 10 million words (*10M dataset*), and (b) 100 million words (*100M dataset*). In this shared task, performance of the sentence filtering is estimated as the translation quality (as measured by BLEU) of phrase-based statistical machine translation (SMT) and neural machine translation (NMT) systems built from the subselected data. Evaluation sets belong to different domains, which discourages strategies based on domain relatedness.

Our submission is built upon the assumption that a training set that maximizes the quality of machine translation (MT) must meet the following requirements:

- Parallel sentences must be mutual translations.
- Sentences must be *fluent* in the corresponding language in order to build a reliable language model/NMT decoder. We work under the hypothesis that the sentence `D0006 Tooth brush A NOELL / F945J 0,21` is less useful for a language model than `I brush my teeth and look in the mirror`, despite containing a similar amount of tokens.
- Vocabulary must be diverse, since the MT systems are evaluated with test sets from different domains.

We built a training corpus that meets the aforementioned requirements in a sequential process that comprises the following steps:

1. As a preprocessing step, deletion of parallel sentences by means of a set of hand-crafted *hard rules* implemented in the translation memory cleaning tool `Bicleaner`.[3] These rules are addressed at detecting evident flaws such as languages different from English and German, encoding errors, very different lengths in parallel sentences, etc. and speeding up the subsequent steps.
2. Detection of misaligned parallel sentences by means of an automatic classifier.
3. Scoring of sentences based on fluency and diversity: four different approaches were tested and submitted.

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

[2] As part of the Paracrawl project: https://paracrawl.eu/.

[3] https://github.com/bitextor/bicleaner

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 955–962
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64115

The remainder of the paper is organized as follows: Section 2 outlines related approaches, Sections 3 and 4 respectively describe the steps 2 and 3 of our processing pipeline. Section 5 confirms the positive impact of our processing pipeline on translation quality by comparing it with other baseline approaches. Finally, the paper ends with some concluding remarks and the suggestion of potential future research directions.

## 2 Related work

The WMT 2018 parallel corpus filtering shared task partially shares its objectives with the First Automatic Translation Memory Cleaning Shared Task (Barbu et al., 2016), where participants had to automatically classify translation memory segments according to whether the target language (TL) side was translation of the source language (SL) side or not. This task is, in turn, very similar to the detection of parallel sentences in comparable corpora, that can be tackled by combining bilingual data and automatic classifiers (Munteanu and Marcu, 2005), machine translation (Abdul-Rauf and Schwenk, 2009) or, more recently, word embeddings (España-Bonet et al., 2017). In fact, the approach we follow to detect sentences that are mutual translations is similar to the work of Munteanu and Marcu (2005). Their approach differs from ours in the fact that we make use of a larger set of shallow features not related to lexical similarity.

However, since the size of the data sets that participants must produce in this task is smaller than the number of parallel sentences that are mutual translations, this task is also related to the *data selection*: selection of a subset of data that maximizes translation quality, avoiding redundancy and matching a given domain (Eetemadi et al., 2015). Instead of the widespread language-model based data selection methods (Axelrod et al., 2011), we replaced words with placeholders in order to not take into account the domain of the text.

## 3 Sentence alignment classifier

After applying the hard rules aimed at detecting evident flaws introduced in Section 1, 22 229 462 parallel sentences (21%) out of the initial 104 002 521 were kept. In order to discard pairs of sentences that are not mutual translations, we applied an automatic classifier to the sentence pairs that passed the hard rule filter. The classifier produces a score for each pair of sentences that represents the probability that they are mutual translations. This score is used in different ways depending on the scoring strategy chosen for achieving vocabulary diversity and fluency (see next section).

The features we used can be split in two groups: those that represent the lexical similarity of the two sides of a parallel sentence by making use of probabilistic bilingual dictionaries, and those that are based on shallow properties such as sentence length, capitalized words, punctuation marks, etc.

Given a bilingual probabilistic dictionary whose SL is *L1* and TL is *L2* and a pair of sentences $(s_1, s_2)$, written in languages *L1* and *L2* respectively, we computed the four lexical similarity features described next. The feature `DICT-QMAX-L1` is defined as $\prod_{w \in s_2} \max_{w' \in s_1} p(w', w)$, where $p(w', w)$ is the translation probability from the *L1* word $w'$ to the *L2* word $w$ according to the bilingual dictionary. That is, `DICT-QMAX-L1` is the product, for each word $w$ in $s_2$, of the maximum translation probability from any word in $s_1$ to $w$. The feature `DICT-QMAX-L2` is computed in the opposite direction (with the help of a bilingual dictionary whose SL is *L2* and TL is *L1*). We also used two additional features that account respectively for the proportion of words in $s_1$ and $s_2$ that can be found in the bilingual dictionaries.

Shallow features include, among others:

- For each language, probability of the sentence length according to a Poisson distribution, given the sentence length ratio observed in the positive examples of the classifier training set.[4]
- Number of tokens in each segment.
- Average token length (in characters) in each segment.
- Number of punctuation marks in each segment.
- Number of numerical expressions in each segment that can be found in the other segment of the pair.

[4] Let $l_s$ be the length of the SL sentence, $l_t$ the length of the TL sentence and $r$ the average length of TL sentence to length of SL sentence ratio observed in the training corpus. The probability of the TL sentence length is computed as $e^{-l_s r} \frac{l_s r}{l_t!}$.

- Number of capitalized tokens in one segment that can be found in the other segment of the pair.

We trained a Random Forest classifier (Breiman, 2001) with 200 trees and a maximum depth of 2. The remaining parameters were the default ones in the Random Forest implementation of the `Scikit-learn` library.[5]

The bilingual dictionaries were obtained from all the available English–German parallel corpora from WMT 2018 news translation shared task (with the exception of 60 000 sentences randomly removed from *news-commentary-v13*, which were used for training the classifier, as explained in the next paragraph). After concatenating the corpora, they were word-aligned by means of `MGIZA++`,[6] alignments were symmetrized with the heuristic *grow-diag-final* and the probabilities in the bilingual dictionaries were estimated by maximum likelihood from the symmetrized alignments. Before building the dictionaries and computing the lexical features, compounds in German were segmented with the maximum entropy classifier proposed by Dyer (2009).[7]

The training set for the classifier was built as follows. From the 60 000 parallel sentences randomly removed from the *news-commentary-v13* parallel corpus, 50 000 were used for actually training the classifier while the remaining 10 000 were used as a validation set. From the training set, 50 000 positive instances were obtained. 50 000 negative instances were also obtained from the training set, after randomly shuffling their English side, i.e., synthetically generating pairs of sentences that are not mutual translations. The same strategy was built for obtaining negative instances for the validation set. The accuracy of the resulting classifier with the score threshold at 0.5 was 0.98.

## 4 Scoring for fluency and diversity

From the three main issues that need to be tackled for obtaining a good training corpus for machine translation, the classifier dealt with sentences that are not mutual translations. In this section, we describe the four scoring strategies we submitted to the shared task and how they tackle the two remaining issues: vocabulary diversity and fluency.

### 4.1 N-gram saturation

This scoring strategy aims to increase the vocabulary diversity by removing sentence pairs that are too similar to other pairs in the training corpus. Each sentence pair is assigned the score returned by the classifier, with the exception of those sentences deemed as too similar, which are discarded. The 10M and 100M datasets are just obtained by selecting the not discarded (not deemed as too similar) sentences, sorted in descending classifier score, until the desired token count is achieved.

Too similar sentences are identified by a simple $n$-gram saturation algorithm. First, some tokens are replaced with placeholders. Fully alphabetic tokens written either in lowercase (all characters are lowercase) or in titlecase (the first character is uppercase and the remaining ones are lowercase) are kept intact and every other token is replaced with one of the following placeholders:

- `ALPHA:UPPER`: all characters are uppercase.

- `ALPHA:MIXED`: all characters are alphabetic, but the token is neither written in lowercase, nor in titlecase, nor in full uppercase.

- `NUMERIC`: all the characters are digits.

- `PUNCTUATION`: all the characters are punctuation marks.

- `MIXED`: none of the previous conditions are met.

Additionally, titlecased words that can be found in the other sentences of the pair are replaced with `ALPHA:PROPER`.

For instance, the sentence `the Kari EL22 electrode switch is designed for the control of conductive liquids .` becomes `the ALPHA:PROPER MIXED electrode switch is designed for the control of conductive liquids PUNCTUATION` after the replacement is made.[8]

Once placeholders are introduced in sentences, sentence pairs are traversed in descending classifier score order, and those whose full set of 4-grams can be found in sentences with higher

[5] `http://scikit-learn.org/`

[6] `https://github.com/moses-smt/mgiza.git`

[7] `https://github.com/redpony/cdec/tree/master/compound-split`; pre-trained models from this implementation were used.

[8] The word `Kari` also appears in the German sentence and it is thus considered as a proper noun.

scores are classified as too similar and discarded. Placeholders prevent sentences which differ from other sentences only in proper nouns, codes, figures, punctuation, etc. from being accepted.

The number of sentences retained after applying $n$-gram saturation was 10 100 275, from which the top 433 760 and the top 5 121 715 with the highest classifier scores were respectively selected to build the 10M and 100M datasets.

### 4.2 Active learning data selection

A potential limitation of the scoring strategy based on $n$-gram saturation is that, when building the 10M word training set, a large proportion of the sentences which passed the saturation filter were not considered. From the 6 798 687 sentences resulting from applying $n$-gram saturation with a classifier score above 0.5 (i.e., very likely to be mutual translations), 433 760 were greedily chosen without even considering the remaining ones. These sentences could contain useful words or expressions that have been ignored.

In order to overcome that limitation, we designed a data selection strategy that considers the vocabulary of the whole corpus. Our approach is an adaptation of the active learning strategy used for building training corpora for SMT proposed by Haffari et al. (2009) and it is outlined in Algorithm 1. This algorithm is applied only to sentences with a classifier score $\geq 0.55$; those below that score are discarded.

**Algorithm 1** Data selection via active learning

```
Require: Bilingual corpus $C$
Ensure: Sorted bilingual corpus $S$
  $S \leftarrow \emptyset$
  $blocksize \leftarrow 100\,000$
  while $|C| > 0$ do
    $S_{new} \leftarrow select(C, S)$
    $C \leftarrow C - S_{new}$
    $S \leftarrow S + S_{new}$
    $blocksize \leftarrow increaseBlockSize(blocksize)$
  end while
```

It iteratively selects a sequence of sentence pairs $S_{new}$ and appends it to the sorted corpus $S$ until no sentences are available in the corpus $C$. The function $select(C, S)$ scores the sentences in $C$ with the *Geom n-gram* function (Haffari et al., 2009, Sec. 3.1.2), sorts them by decreasing score, applies the $n$-gram saturation filter described previously (with a small modification: a sentence pair is discarded if at least half of the 4-grams have been observed in not discarded sentence pairs from $C$ with higher score) and returns the top $blocksize$ sentences. The *Geom n-gram* scoring function assigns the highest scores to sentences with $n$-grams that are frequent in $C$ and infrequent in $S$. The function *increaseBlockSize* doubles the block size every 5 iterations. The datasets were built by traversing the sorted corpus $S$ until desired token counts were achieved.

### 4.3 Language modeling

While the two previous approaches aimed at increasing the diversity of the vocabulary, the corpora selected following these approaches may contain pairs of sentences that are not useful to build a powerful language model, such as: `Brush for Acrylic - blue #06` ↔ `Pinsel für Acryl Falten - Rot #6`.

In order to include only *fluent* sentences in the training sets, we made use of language models. As we did not want to include a bias towards news data in the language models, placeholders were used in a similar way to what has been described in Section 4.1. The following types of tokens were replaced with placeholders:

- Tokens made fully of alphabetical characters. They were replaced with a placeholder that represents its capitalization: lowercase (`ALPHA:LOWER`), titlecase (`ALPHA:TITLE`), uppercase (`ALPHA:UPPER`) or mixed case (`ALPHA:MIXED`).
- Tokens made fully of numeric characters (`ALPHA:NUM`).
- Tokens that contain a numeric or alphabetical character but do not fall into any of the two previous groups (`MIXED`).

Consequently, tokens made only of punctuation characters were kept unchanged. The previous pair of sentences was hence processed as follows: `ALPHA:TITLE ALPHA:LOWER ALPHA:TITLE - ALPHA:LOWER MIXED` ↔ `ALPHA:TITLE ALPHA:LOWER ALPHA:TITLE ALPHA:TITLE - ALPHA:TITLE MIXED`

Each 5-gram language model (one for each language) was estimated from 20 000 000 sentences randomly chosen from the news and Europarl monolingual corpora with KenLM (Heafield, 2011) and Knesser-Ney smoothing (Heafield et al., 2013).

Language models were used to score pairs of sentences as follows:

1. Pairs of sentences with a classifier score lower than 0.55 were discarded.
2. Remaining pairs of sentences were sorted in ascending sum of (English plus German) perplexity per word.
3. The $n$-gram saturation algorithm described in Section 4.1 was applied. As similar sentences have similar perplexities, the algorithm is needed in order to decrease the degree of repetition in the resulting corpus.

Two submissions were based on language model scoring. In the first one, `prompsit-lm`, sentences were truecased before training the language model and the saturation algorithm was applied exactly as described in Section 4.1, i.e. with the same placeholder replacement strategy. In the alternative submission, `prompsit-lm-nota`, sentences were not truecased for language model scoring and the saturation algorithm was applied without placeholder replacement.[9]

In the submission `prompsit-lm`, 5 868 776 sentences passed the $n$-gram saturation filter, from which the 4 492 314 sentence pairs with the lowest perplexity per word were selected for building the 100M tokens training set. In the submission `prompsit-lm-nota`, since the saturation filter is less aggressive, 7 016 169 sentence pairs passed that filter and 4 491 269 were selected for the 100M tokens training set.

## 5 Machine translation experiments

We built MT systems from the four scoring alternatives presented and compared them with two baseline systems: one in which the sentences were randomly chosen from the noisy, crawled data (`random`) and another one in which the hard-rule filtering was applied and each sentence was simply scored by the classifier (`only-classifier`; 10M and 100M datasets were built by selecting sentences in descending classifier score order).

Systems were trained following the official instructions from the shared task.[10] SMT systems were built with Moses and tuned with Batch MIRA (Cherry and Foster, 2012). A 5-gram language model was estimated from the TL side of the training corpus. NMT systems followed the Transformer architecture (Vaswani et al., 2017) and were built with Marian (Junczys-Dowmunt et al., 2018). 49 500 byte pair encoding merge operations (Sennrich et al., 2016) were applied to segment the words in the NMT training corpus. The development set (used for tuning the parameters of the log-linear model in SMT and for early stopping in NMT) was *newstest2016*, while the test set was *newstest2017*. Table 1 presents the (cased) BLEU scores obtained by the MT systems built.

It can be observed that the scores of NMT systems trained on random subsamples (`random` baseline) are very low if we compare them with SMT. This confirms that NMT is very sensitive to noisy training data (Belinkov and Bisk, 2017). An important increase in BLEU for all systems can be observed when filtering with hard rules and classifier (`only-classifier` system). After this filtering, NMT outperforms SMT for both training set sizes.

Concerning our submissions, results show that adding $n$-gram saturation (`prompsit-sat`) slightly improves the results in the four datasets, which confirms that vocabulary diversity is relevant for this task. We can also observe in Table 3 that the number of unknown words in the test set was slightly reduced. Our active learning strategy for achieving vocabulary diversity (`prompsit-al`), however, brought a degradation in the 10M dataset and a light improvement in the 100M one. If we analyze vocabulary sizes (displayed in Table 2), it was reduced (in comparison with `prompsit-sat`) only for the 10M dataset, and the number of unknown words in the test set increased. A potential solution for this issue could be reducing the block size for the first iterations of the active learning algorithm, so that more itera-

[9] Note that, in the `prompsit-lm` submission, two different placeholders replacement strategies were applied. Firstly, that described in Section 4.3 was applied in order to obtain language model perplexities. Afterwards, the one described in Section 4.1 was applied in order to discard similar sentences. In the `prompsit-lm-nota` submission, only the first one was applied. Concerning truecasing, preliminary experiments showed that it has a limited impact for language model scoring, hence the main difference between the submissions is the strength of $n$-gram saturation: fewer sentences are discarded if placeholders are disabled.

[10] `http://www.statmt.org/wmt18/parallel-corpus-filtering.html`

| System | SMT 10M | SMT 100M | NMT 10M | NMT 100M |
|---|---|---|---|---|
| random | 14.92 | 18.51 | 7.70 | 7.66 |
| only-classifier | 20.22 | 23.96 | 21.46 | 29.32 |
| prompsit-sat | 20.77 | 24.12 | 22.82 | 29.55 |
| prompsit-al | 20.02 | 24.46 | 22.50 | 29.64 |
| prompsit-lm | 19.09 | 24.37 | 18.50 | 29.79 |
| prompsit-lm-nota | 18.61 | 24.36 | 18.60 | 29.85 |

Table 1: BLEU scores obtained by our 4 submissions and two baseline approaches.

tions are executed before obtaining the 10M training set.

The submissions that aimed at increasing the fluency of the training corpus brought a light improvement in translation quality for the NMT system trained on the 100M dataset. On the contrary, they further reduced the vocabulary sizes and increased the unknown rate for the 10M dataset. We believe this is due to the fact that, with this approach, fluency had a stronger influence than vocabulary diversity in the criterion for selecting sentences for the small dataset. Only the top 836 520 sentences with smallest perplexity were explored for building the final 10M training corpus obtained with `prompsit-lm`, which contained 551 098 sentences.[11] A manual inspection of the sentences included in the 100M dataset but not in the 10M one showed that they were perfectly fluent. This means fluent sentences which are more interesting (from a vocabulary point of view) have been ignored when building the 10M dataset, since the process is mainly guided by perplexity. This problem disappears in the large data set, that is large enough to contain diverse vocabulary.

The BLEU scores reported in this section do not exactly match those published in the official results (Koehn et al., 2018) because, unlike the scores reported in this paper, the official scores were averaged over multiple training runs and multiple evaluation corpora. Nevertheless, the relative performance of our four submissions remains the same. Our active learning and language model scoring strategies were very competitive for the 100M dataset and were ranked very close to the top performing systems, while our best performing submissions for the 10M dataset were in the middle of the ranking.

## 6 Concluding remarks

This paper described Prompsit Language Engineering's submissions to the WMT 2018 parallel corpus filtering shared task. Our four submissions stemmed from a strategy based on handcrafted filtering rules and an automatic classifier that selects those sentences that are mutual translations. Our submissions explored different ways of achieving vocabulary diversity and fluency in the selected training corpora. The strategies based on an active learning algorithm (aimed at achieving vocabulary diversity) and language model perplexity combined with $n$-gram saturation (aimed at achieving fluency and vocabulary diversity) allowed our submissions to be ranked close to the top performing system for the 100M dataset.

Our strategies were less successful for the 10M tasks, as they were placed in the middle of the ranking. An analysis of out of vocabulary words in the test set for the language model-based approaches suggests that fluency has a stronger influence than vocabulary diversity. A scoring scheme that balances them better should improve the results and designing it could be a future research direction. The active learning algorithm could also be tuned for smaller datasets by decreasing the block size parameter.

## Acknowledgments

We would like to thank the anonymous reviewers for their suggestions on how to improve the paper. Work funded by EU project 2016-EU-IA-0114 "Provision of web-scale parallel corpora for official European languages".

## References

Sadaf Abdul-Rauf and Holger Schwenk. 2009. On the use of comparable corpora to improve smt performance. In *Proceedings of the 12th Conference of the*

[11] The difference between these two numbers is the amount of sentences removed by the $n$-gram saturation algorithm.

| System | de 10M | en 10M | de 100M | en 100M |
|---|---|---|---|---|
| random | 904K | 789K | 3 810K | 3 364K |
| only-classifier | 561K | 382K | 2 197K | 1 274K |
| prompsit-sat | 585K | 365K | 2 246K | 1 174K |
| prompsit-al | 403K | 228K | 2 329K | 1 162K |
| prompsit-lm | 359K | 99K | 2 022K | 910K |
| prompsit-lm-nota | 364K | 103K | 1 969K | 879K |

Table 2: Vocabulary sizes, expressed in thousands of words, after tokenization with the Moses tokenizer, of the training corpora produced with our four submissions and two baseline approaches.

| System | # unks 10M | # types 10M | # unks 100M | # types 100M |
|---|---|---|---|---|
| random | 2 580 | 2 012 | 1 132 | 913 |
| only-classifier | 2 852 | 2 207 | 1 199 | 921 |
| prompsit-sat | 2 639 | 2 027 | 1 148 | 877 |
| prompsit-al | 3 084 | 2 266 | 1 114 | 848 |
| prompsit-lm | 4 307 | 2 744 | 1 178 | 882 |
| prompsit-lm-nota | 4 183 | 2 682 | 1 182 | 896 |

Table 3: Unknown words in the source language (German) size of the *newstest2017* test set. The columns labeled as `# unks` represent the number of instances of unknown words, while `# types` stands for the number of unique unknown words.

*European Chapter of the Association for Computational Linguistics*, EACL '09, pages 16–23, Stroudsburg, PA, USA. Association for Computational Linguistics.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eduard Barbu, Carla Parra Escartín, Luisa Bentivogli, Matteo Negri, Marco Turchi, Constantin Orasan, and Marcello Federico. 2016. The first automatic translation memory cleaning shared task. *Machine Translation*, 30(3):145–166.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Colin Cherry and George Foster. 2012. 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*, NAACL HLT '12, pages 427–436, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chris Dyer. 2009. Using a maximum entropy model to build segmentation lattices for mt. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 406–414, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sauleh Eetemadi, William Lewis, Kristina Toutanova, and Hayder Radha. 2015. Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29(3):189–223.

Cristina España-Bonet, Ádám Csaba Varga, Alberto Barrón-Cedeño, and Joseph van Genabith. 2017. An empirical analysis of nmt-derived interlingual embeddings and their use in parallel sentence identification. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1340–1350.

Gholamreza Haffari, Maxim Roy, and Anoop Sarkar. 2009. Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 415–423, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.*, 31(4):477–504.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

# NICT's Corpus Filtering Systems for the WMT18 Parallel Corpus Filtering Task

Rui Wang Benjamin Marie* Masao Utiyama Eiichiro Sumita
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Souraku-gun, Kyoto, 619-0289, Japan
{wangrui, bmarie, mutiyama, eiichiro.sumita}@nict.go.jp

## Abstract

This paper presents the NICT's participation in the WMT18 shared parallel corpus filtering task. The organizers provided 1 billion words German-English corpus crawled from the web as part of the Paracrawl project. This corpus is too noisy to build an acceptable neural machine translation (NMT) system. Using the clean data of the WMT18 shared news translation task, we designed several features and trained a classifier to score each sentence pairs in the noisy data. Finally, we sampled 100 million and 10 million words and built corresponding NMT systems. Empirical results show that our NMT systems trained on sampled data achieve promising performance.

## 1 Introduction

This paper describes the corpus filtering system built for the participation of the National Institute of Information and Communications Technology (NICT) to the WMT18 shared parallel corpus filtering task.

NMT has shown large gains in quality over Statistical machine translation (SMT) and set several new benchmarks (Bojar et al., 2017). However, NMT is much more sensitive to domain (Wang et al., 2017) and noise (Khayrallah and Koehn, 2018). The reason is that NMT is a single neural network structure, which would be affected by each instance during the training procedure (Wang et al., 2017). In comparison, SMT is a combination of distributed models, such as a phrase-table and a language model. Even if some instances in the phrase-table or the language model are noisy, they can only affect part of the models and would not affect the entire system so much. To the best of our knowledge, there are only few works investigating the impact of the noise problem in NMT (Xu and Koehn, 2017; Belinkov and Bisk, 2017).

In this paper, we focus on the performance of NMT trained on noisy parallel data. We adopt the clean data of WMT18 News Translation Task to train a classifier and compute informative features. Using this classifier, we score each sentence in the noisy data and sample the top ranked sentences to construct the pseudo clean data. The new pseudo clean data are used to train a robust NMT system.

The remainder of this paper is organized as follows. In Section 2, we introduce the task and data. In Section 3, we introduce the features that we designed to score sentences in the noisy corpus. We use these features to train a classifier and the sentences in the noisy corpus are scored by this classifier. Empirical results produced with our systems are showed and analyzed in Section 4, and Section 5 concludes this paper.

## 2 Task Description

WMT18 shared parallel corpus filtering task[1] (Koehn et al., 2018) provides a very noisy 1 billion words (English word count) German-English (De-En) corpus crawled from the web as a part of the Paracrawl project. Participants are asked to provide a quality score for each sentence pair in the corpus. Computed scores are then evaluated given the performance of SMT and NMT systems trained on 100M and 10M words sampled from data using the quality scores computed by the participants. *newstest2016* is used as the development data and the test data include *newstest2018*, *iwslt2017*, *Acquis*, *EMEA*, *Global Voices*, and *KDE*.[2] The statistics of the noisy data to filter are shown in Table 1.

The participants may use the WMT18 News

[*] The first two authors have equal contributions.

[1] http://www.statmt.org/wmt18/parallel-corpus-filtering.html

[2] Note that, except for *newstest2018*, all testsets remained unknown from the participants until the submission deadline.

*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 963–967
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64116

| Language | #lines | #words | #tokens |
|---|---|---|---|
| En | 104.00 M | 1.00B | 1.66B |
| De | 104.00 M | 0.96B | 1.62B |

Table 1: Statistics of the noisy data to filter. "#words" indicates the word count before tokenization.

Translation Task data[3] for German-English (without the Paracrawl parallel corpus) to train components of their method. In addition, to participate in the shared task, participants have to submit a file with quality scores, one score per line, corresponding to the sentence pairs. The scores do not have to be meaningful, except that higher scores indicate better quality.

## 3 Sentence Pairs Scoring

The task requires to give a score to each sentence pair in the corpus to filter. We performed first an aggressive filtering (Section 3.1) to avoid scoring sentence pairs that are clearly too noisy to be used during the training of MT systems. Then, we computed informative features (Section 3.2) for each one of the remaining sentence pairs. Then, according to the feature scores, a classifier computes a global score for each sentence pair that can be used to rank them.

### 3.1 Aggressive Filtering

After a quick observation of the data, we first decided to perform an aggressive filtering since it appeared that many of the sentence pairs are obviously too noisy to be used to train MT systems. For instance, many sentences in the corpus are made of long sequences of numbers or punctuation marks. We decided to give a score of 0.0 to all the sentence pairs that contain a sentence made of tokens that are, for more than 25% them, numbers or punctuation marks. We also had to take into account the sentence length: very short source sentences are more likely to be paired with a good translation in the corpus, and our classifier may give to such pairs very high scores. Then, in order to avoid a filtering that keeps sentences made in majority of very short and redundant sentences, that are not very useful to train NMT systems, we also give a score of 0.0 to all sentence pairs that contain a source or a target sentence that contains less than four tokens. We also give a score of 0.0 to all the sentence pairs that contain a sentence longer than 80 tokens since the default parameters of the SMT system used for evaluation filter out sentences longer than that.

This aggressive filtering excluded 69% of the sentence pairs, leaving us a much reduced quantity of sentence pairs to be scored by our classifier.

### 3.2 Features

We scored each of the remaining sentence pairs with four NMT transformer models, trained with Marian (Junczys-Dowmunt et al., 2018)[4], on all the parallel data provided for the shared news translation task (excluding the "paracrawl" corpus). We trained left-to-right and right-to-left models for German-to-English and English-to-German translation directions. We used these four model scores as features in our classifier.

We also trained lexical translation probability with Moses and used them to compute a sentence-level translation probability, for both translation directions, as proposed by Marie and Fujita (2017).

To evaluate the semantic similarity between the source and target sentence, we compute a feature based on bilingual word embeddings as follows. First, we trained monolingual word embeddings with FastText (Bojanowski et al., 2017)[5] on the monolingual English and German data provided by the WMT organizers. Then, we aligned English and German monolingual word embedding spaces in a bilingual space using the unsupervised method proposed by Artetxe et al. (2018).[6] Given the bilingual word embeddings, we computed embeddings for the source and target sentence by doing the element-wise addition of the bilingual embedding of the words they contain. Finally, we computed the cosine similarity between the embeddings of source and target sentence for each sentence pair, and used it as a feature.

Other features are computed to take into account the sentence length: the number of tokens in the source and target sentences, and the difference, and its absolute value, between them. We summarize the features that we used in Table 2.

[3] http://www.statmt.org/wmt18/translation-task.html

[4] https://marian-nmt.github.io/

[5] We used the default parameters for skipgram, with 512 dimensions.

[6] We used the implementation provided by the authors, with default parameters, at: https://github.com/artetxem/vecmap.

| Feature | Description |
|---|---|
| L2R (2) | Scores given by the left-to-right German-to-English and English-to-German NMT models |
| R2L (2) | Scores given by the right-to-left German-to-English and English-to-German NMT models |
| LEX (4) | Lexical translation probabilities, for both translation directions |
| WE (1) | Bilingual sentence embedding similarity |
| LEN (4) | Length-based features |

Table 2: Set of features used by our classifier.

### 3.3 Classifier

We chose a logistic regression classifier to compute a score for each sentence pair using the features presented in Section 3.2. We trained our classifier on *Newstest2014*, that we used as positive examples of good sentence pairs, and created the same number of negative examples using the following procedure. We created three-type of negative examples, each of which contains one third of the sentence number of *Newstest2014*:

- Misaligned: The target sentences are wrongly aligned to the previous or following source sentences.
- Wrong translation: some words in a sentence are replaced by random words from the vocabulary.
- Misordered words: we shuffled the words in a sentence.

We used the same procedure to create training data with *Newstest2015*, and used it to tune the regularization parameter of our classifier. The classifier accuracy is 78.9% on *Newstest2015*.

We used the probability returned by the classifier for each sentence pair as the score to be used to perform filtering.

## 4 NMT Systems and Results

For this task, we did not conduct experiments with a state-of-the-art NMT system, because the organizers fixed the data and systems settings for a fair comparison.

### 4.1 NMT Systems

For the data preprocessing, we strictly followed the data preparation (including tokenization, truecasing, and byte pair encoding) provided by the organizers. To train NMT systems, we used the provided official settings of Marian, which can be found at the WMT offical website[7] and the Appendix A. All our NMT systems were trained on four Nvidia Tesla P100 GPUs.

Our settings were the same for all of the NMT systems. For each method, we use their score to select the top 100M and 10M sentences to train the corresponding NMT systems. In Table 4, “Original” means the original corpus without any filtering. “Aggressive Filtering” is the method which we introduced in Section 3.1. “Hunalign” indicates the baseline corpus filtering method (Varga et al., 2007)[8] given by the organizers. “Classifier” indicates the classifier that we proposed in Section 3.3. “Classifier + LangID” indicates that we also use a language identification tool, LangID (Lui and Baldwin, 2012)[9], to filter the sentence pairs containing sentences that are not German or English. The results were evaluated on the development data *newstest2016*.

### 4.2 NMT Performance

From the results in Table 4, we have the following observations:

- The proposed “Aggressive Filtering” reduced 69% sentences and improved 1.5 BLEU compared to using the original corpus. This indicates that most of the noisy data can be filtered by the aggressive filter.
- The baseline “Hunalign” did not perform very well, the performance decreased to 3.6/0.03 by selecting 100/10M sentences. Especially when selecting 10M sentences, the NMT system nearly did not work.
- The proposed “Classifier” significantly improved NMT performance by more than 20

[7] http://www.statmt.org/wmt18/parallel-corpus-filtering-data/dev-tools.tgz

[8] http://mokk.bme.hu/resources/hunalign/

[9] https://github.com/saffsd/langid.py

| System | newstest2018 | iwslt2017 | Acquis | EMEA | Global Voices | KDE | average |
|---|---|---|---|---|---|---|---|
| SMT-10M | 27.79 | 20.94 | 19.27 | 25.89 | 21.38 | 25.51 | 23.46 |
| SMT-100M | 30.79 | 22.76 | 21.98 | 30.39 | 23.63 | 26.55 | 25.98 |
| NMT-10M | 32.93 | 23.67 | 21.67 | 27.60 | 25.13 | 24.65 | 25.94 |
| NMT-100M | 37.28 | 25.83 | 26.11 | 34.13 | 27.62 | 29.25 | 30.04 |

Table 3: WMT official results.

| Methods | #tokens (En) | #lines | #BLEU |
|---|---|---|---|
| Original | 1.6B | 104.0M | 7.4 |
| Aggressive Filtering | 584M | 31.9M | 8.8 |
| Hunalign | 100M | 8.7M | 3.6 |
| Classifier | 100M | 9.1M | 26.1 |
| Classifier + LangID | 100M | 6.7M | 31.6 |
| Hunalign | 10M | 2.6M | 0.03 |
| Classifier | 10M | 1.2M | 25.6 |
| Classifier + LangID | 10M | 0.9M | 27.8 |

Table 4: Results on the development data.

| Methods | #tokens (En) | #Time |
|---|---|---|
| Original | 1.6B | 43 hours |
| Aggressive Filtering | 584M | 47 hours |
| Classifier + LangID | 100M | 55 hours |
| Classifier + LangID | 10M | 11 hours |

Table 5: Training efficiency.

BLEU. This indicates that the proposed classifier can rank sentence by a proper order and the more useful sentences are selected.

- The "Classifier + LangID" achieved further approximately 2∼5 BLEU improvement. This indicates there are several sentences which are not proper languages and they can be detected by the LangID.

- For the proposed method, the systems built from 100M sentences performed much better than the ones built from 10M sentences. This indicates that filtering too many sentences will harm the NMT performance.

### 4.3 Training Efficiency

Besides the NMT performances, we also showed the training efficiency in Table 5.

The results in Table 5 showed:

- The training time of using 1.6B, 584M, and 100M sentences was very close.

- The training time of using 10M sentences was quite faster than the other ones. Together with the performance results in Table 4, it show that these 10M contains most of the useful information in the entire corpus and can accelerate NMT training significantly.

### 4.4 Official Results

We reported the official results of our submitted system "Classifier + LangID" in Tables 3. In the official results, both SMT and NMT results were reported.

From the results in Table 3, we have the following observations:

- The NMT system performed much better than corresponding SMT systems. This indicates that the proposed method can help NMT in overcoming the noise problem.

- The systems built from 100M sentences performed much better than the ones built from 10M sentences. This is consistent with the results obtained on the development data.

- Compared with other teams, the rankng of our SMT systems performed better than our NMT systems. The reason may be that we used several features from SMT. We ranked the first in the KDE SMT-10M task.

## 5 Conclusion and Future Work

In this paper, we investigated the noisy data problem in NMT. We designed a classification system to filter the noisy data for the WMT18 shared parallel corpus filtering task and built NMT systems using the selected data.

The empirical results showed that most of the sentence pairs in the corpus are noisy. By removing these sentence pairs, the training corpus can be reduced up to 1% of the original one while training a significantly better NMT system than the original NMT system trained on all the data. In our future work, we would like to investigate the impact of each type of noise and the effect of each feature used by our classifier.

In this paper, we focused on supervised classification methods. That is, we used clean data as a gold standard. In our future work, we would like to investigate this task using unsupervised methods. That is, we only use the noisy data and let NMT itself detect noisy sentence pairs.

## Acknowledgments

This work is partially supported by the program "Promotion of Global Communications Plan: Research, Development, and Social Demonstration of Multilingual Speech Translation Technology" of MIC, Japan.

## A Marian Settings

To train NMT systems, we used the provided settings of Marian:

```
--sync-sgd -T --devices 0
1 2 3 --mini-batch-fit -w
3000 --dim-vocabs 50000
50000 --layer-normalization
--dropout-rnn 0.2 --dropout-src
0.1 --dropout-trg 0.1
--learn-rate 0.0001
--after-epochs 0 --early-stopping
5 --max-length 80 --valid-freq
20000 --save-freq
20000 --disp-freq 2000
--valid-mini-batch 8
--valid-metrics cross-entropy
perplexity translation --seed
1111 --exponential-smoothing
--normalize=1 --beam-size=12
--quiet-translation.
```

## References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, and Julia Kreutzer. 2017. Proceedings of the second conference on machine translation. In *Proceedings of the Second Conference on Machine Translation*. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in c++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Benjamin Marie and Atsushi Fujita. 2017. Efficient extraction of pseudo-parallel sentences from raw monolingual data using word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 392–398. Association for Computational Linguistics.

Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, 292:247.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488, Copenhagen, Denmark. Association for Computational Linguistics.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950. Association for Computational Linguistics.

# Author Index